

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2001 年 5 月 17 日 (17.05.2001)

PCT

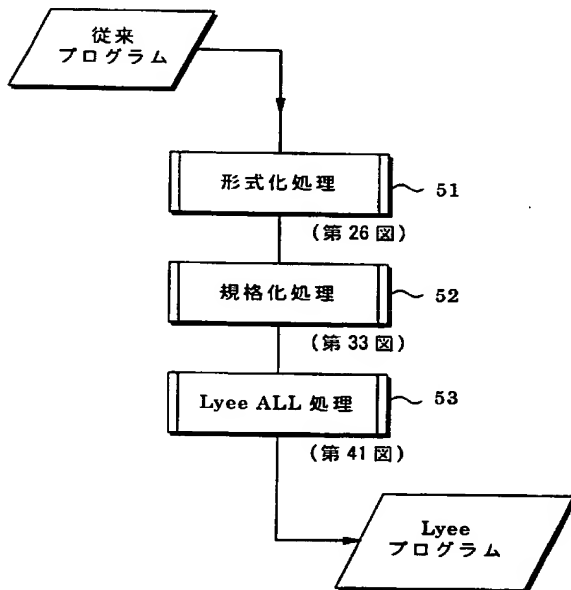
(10) 国際公開番号
WO 01/35213 A1

- (51) 国際特許分類⁷: G06F 9/44 (71) 出願人 (米国を除く全ての指定国について): 株式会社 アイエスデー研究所 (INFORMATION SYSTEM DEVELOPMENT INSTITUTE) [JP/JP]; 〒108-0074 東京都港区高輪三丁目11番3号 Tokyo (JP). ソフトウェア生産技術研究所株式会社 (THE INSTITUTE OF COMPUTER BASED SOFTWARE METHODOLOGY AND TECHNOLOGY) [JP/JP]; 〒108-0074 東京都港区高輪三丁目11番3号 Tokyo (JP).
- (21) 国際出願番号: PCT/JP00/07916
- (22) 国際出願日: 2000 年 11 月 10 日 (10.11.2000)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願平 11/321788
1999 年 11 月 11 日 (11.11.1999) JP
- (72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 根来文生 (NEGORO, Fumio) [JP/JP]; 〒248-0001 神奈川県鎌倉市十二所 967-64 Kanagawa (JP).

[続葉有]

(54) Title: PROGRAM REPRODUCING METHOD AND DEVICE, AND MEDIUM ON WHICH A PROGRAM FOR PROGRAM REPRODUCTION RECORDING

(54) 発明の名称: プログラム再生方法および装置ならびにプログラム再生のためのプログラムを記録した媒体



(57) Abstract: A word used in an existing program is extracted from the program, with a defined object relating to the word identified. The extracted word is assigned to one or more palettes (W04, W02, W03) according to the type of the defined object. An instruction statement or an instruction in the existing program is converted into a form complying with one of a plurality of program constituents in the palette or palettes to which the word is assigned and included in the instruction statement or instruction.

- 51...FORMALIZATION
52...NORMALIZATION
53...Lyee ALL PROCESSING
A...EXISTING PROGRAM
B...Lyee PROGRAM
C...FIG. 26
D...FIG. 33
E...FIG. 41

[続葉有]



(74) 代理人: 弁理士 牛久健司, 外(USHIKU, Kenji et al.);
〒105-0004 東京都港区新橋三丁目4番5号 新橋フ
ンティアビルディング7階 Tokyo (JP).

添付公開書類:
— 国際調査報告書

(81) 指定国 (国内): AU, CA, CN, IN, JP, KR, NZ, RU, SG,
US.

2文字コード及び他の略語については、定期発行される
各PCTガゼットの巻頭に掲載されている「コードと略語
のガイダンスノート」を参照。

(84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE,
DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

(57) 要約:

既存のプログラムから、そこで用いられている単語を、その単語が関連する定義体を識別した形で抽出する。抽出した単語を、その単語が関連する定義体の種類に応じて、一または複数のパレット (W 0 4 , W 0 2 , W 0 3) に割付ける。既存のプログラムの各命令文または命令を、その命令文または命令に含まれる単語が割付けられた一または複数のパレットにおける複数のプログラム構成要素のいずれかに従う形に変換する。

明 細 書

プログラム再生方法および装置ならびに
プログラム再生のためのプログラムを
記録した媒体

技術分野

この発明はプログラム再生方法および装置ならびにプログラム再生のためのプログラムを記録した媒体に関し、さらに詳しくは、Lyeeの開発手法以外の方法により作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeeの開発手法に従う構造をもつプログラムに再構築する方法および装置ならびにそのような処理をコンピュータに実行させるプログラム記録媒体に関する。

背景技術

Lyee (governmental methodology for software providence) のプログラムの開発手法は、従来のソフトウェア作成方法を変革するものとして登場し、開発期間の飛躍的短縮、開発工数の大幅な減少、保守の容易さ、ドキュメント量の現象等の成果をあげるものとして注目されている。

従来のソフトウェア作成方法によって作成された既

存のプログラムでは，システム・エンジニアやプログラマの個々人の個性がプログラム中に現れやすい。このため，既存のプログラムを解析するのは，たとえサポート・ドキュメントがあったとしても容易でないどころか，不可能に近い場合もある。プログラム中に新たな記述を加えることにより新たな機能を追加しようとしても，プログラム解析が容易に行うことができないので新たな機能の追加作業に大変な労力が必要とされることがあった。

さらに，プログラムの動作環境を変更しようとする場合（コンピュータ・マシンやOSの変更等）には，既存のプログラムを一切利用せず，新たにプログラムを作成する以外に方法がないというのが現実であった。

発明の開示

この発明は，既存のプログラムを解析し，その解析結果に基づいて既存のプログラムと同等の機能を達成するプログラムに生まれ変わらせる方法を提供するものである。再生されるプログラムはLyeeの方法論に従うものであり，この発明は再生の処理を自動化できるようにするものである。

この発明による方法は，Lyeeの開発手法以外の方法で作成された既存のプログラムを，上記既存のプログ

ラムが達成する機能を維持して、Lyeの開発手法に従う構造を持つ新たなプログラムに再生する方法である。この方法によると、上記既存のプログラムから、そこで用いられている単語を、その単語が関連する定義体（単語が属する画面、帳票、ファイル、データベース等）を識別した形で抽出し、抽出した単語を、その単語が関連する定義体の種類に応じて、一または複数のパレットに割付け、既存のプログラムの各命令文または命令を、その命令文または命令に含まれる単語が割付けられた一または複数のパレットにおけるLyeの開発手法に従う構造を持つ複数のプログラム構成要素のいずれかに従う形に変換する。

後に詳述するように、Lyeの方法論に従うプログラム（プログラム構成要素）は、W04、W02、W03パレットごとに、単語を基本として設けられるものである。この発明によると、既存のプログラムに含まれるすべての単語を抽出しているから、再生されるプログラムは単語を基本とした（複数）のプログラムの総体であるLyeプログラムの構造を満たすものとなる。既存のプログラムに含まれる単語は、その単語が関連する定義体を識別した形で抽出され、抽出された単語が関連する定義体の種類の応じて一または複数のパレットに割付けられる。そして、既存のプログラムの各命令文または命令については、その命令文または命

令に含まれる単語が割付けられた一または複数のパレットにおける複数のプログラム構成要素（Lyeの方法論に従うプログラム）に対応づけられて変換される。既存のプログラムの命令文または命令が再生されたプログラムにおいても用いられるので、再生されたプログラムは、既存のプログラムと同じ機能を達成する。

この発明によると、既存のプログラムをLyeのプログラムの形に比較的容易にかつほぼ自動的に再構築できる。したがって、既存のコンピュータに搭載されている既存のプログラムを、新たなコンピュータに、同じ機能を達成する新たなプログラムとして比較的容易に移植することが可能となり、プログラム動作環境の変化に対応できる。再生されるプログラムはLyeの方法論に従うものであるから、保守等も容易になる。

プログラム再生方法を、次のように表現することができる。すなわち、この発明による方法は、Lyeの開発手法以外の方法で作成された既存のプログラムを、その既存のプログラムが達成する機能を維持して、Lyeの開発手法に従う構造を持つ新たなプログラムに再生する方法であり、既存のプログラムから既存のプログラムによって制御される定義体を抽出し、抽出した定義体について、W02パレット、W04パレットおよびW03パレットの組から構成される基本構造を動作させるための複数のプログラム構成要素を用意し、

上記既存のプログラムから既存のプログラムで用いられている単語を，その単語が属する定義体およびその単語の入出力区分を識別した形で抽出し，抽出した単語が属する定義体および入出力区分に応じて，その単語を，その単語が属する定義体についてのW02パレット，W04パレットおよびW03パレットのうちの—または複数のパレットに割付け，既存のプログラムの単語を含む命令文または命令を，その命令文または命令に単語を通して関連するパレットにおけるLyeの開発手法に従う構造を持つ複数のプログラム構成要素のうちのいずれかに従う形に変換する。

既存のプログラムには，一般に，そのプログラムによって制御される定義体（画面，ファイル等）の記述が含まれている。定義体についての記述から既存のプログラムによって制御される定義体が把握され，その定義体についての基本構造を動作させるための複数のプログラム構成要素（テンプレート）が用意される。基本構造は，Lyeの方法論に従うプログラム構造であって，W02パレット（プログラム），W04パレット（プログラム）およびW03パレット（プログラム）を含む。

既存のプログラムの記述に基づいて，既存のプログラムにおいて用いられている単語が属する定義体の種類および入出力区分を把握することができる。定義体

の種類および入出力区分に基づいて，既存のプログラムに用いられている単語がW 0 2パレット，W 0 4パレット，W 0 3パレットのいずれに割付けるべき単語であるかが把握される。

既存のプログラムには，定義体のいずれにも属しない単語が用いられていることがある。一般に，既存のプログラムを作成したプログラマが計算途中のデータを格納するために作成した単語であることが多い。このような単語は，その単語を扱うための基本構造（基本構造を動作させるための複数のプログラム構成要素）を別途用意する。

また既存のプログラムにエラー処理の記述が含まれていることがある。この場合には，そのエラー処理を取扱うための基本構造（複数のプログラム構成要素）を別途用意する。

一実施態様では，既存のプログラムの命令文または命令のうち，単語によって表されるフィールドに格納されるべきデータを生成するための命令文または命令を，上記複数のプログラム構成要素のうちデータ生成のためのプログラム構成要素（論理要素プログラム）に従う形に変換する。この命令文または命令は，いわゆる端点単語にデータを格納するための命令文または命令である。端点単語が属する（割付けられている）パレットにおけるデータ生成のためのプログラム構成

要素（論理要素プログラム）において，既存のプログラム中のデータを生成するための命令文または命令が用いられる。

既存のプログラムに用いられている単語が端点単語であるか，または始点単語であるかを識別しておけば，既存のプログラムからの上述の命令文の抽出は容易になる。一般に端点単語や始点単語が用いられている命令文は，命令文に含まれている特定の命令（コマンド）から把握できる。命令文の記述の取決め（プログラム言語ごとの取決め）から，そのような命令文に含まれている２以上の単語のうち，いずれの単語が端点単語であるか，いずれの単語が始点単語であるかを把握することができる。

好ましくは，既存のプログラムの命令文または命令のうち，２以上の単語を含む命令文または命令について，その命令文または命令に用いられている２以上の単語のそれぞれが割付けられている２以上のパレットの関係が，所定の順序を満たす関係であるかどうかを判定する。そして，２以上のパレットの関係が所定の順序を満たしていない場合に，その命令文または命令を複数の命令文または命令に分解し，分解された命令文または命令のそれぞれに新たな単語を含ませる。分解された命令文または命令のそれぞれに含まれる２以上の単語が割付けられている２以上のパレットの関係

が所定の順序を満たすように，新たな単語をW04パレット，W02パレットおよびW03パレットのうち一または複数のパレットに割付ける。

一般に既存のプログラムは，Lyeの開発手法に従ってプログラムが作成されていないので，既存のプログラム中の命令文または命令には，Lyeの開発手法に従って作成されるプログラム中には存在しえない命令文または命令，すなわち，2以上の単語が用いられている命令文または命令であって，その命令文または命令に用いられている2以上の単語のそれぞれが割付けられているパレットの関係が，Lyeの方法論において要求される所定の順序（循環構造）を満たす関係にならないような命令文または命令が記述されていることがある。このような命令文または命令は，上記循環構造の順序を満たすように，その命令文または命令を分解する（1つの命令文を2以上の命令文に書き換える）。既存のプログラム中には循環構造の順序を満たすような命令文または命令が残るので，各命令文または命令のプログラム構成要素のいずれかの形への変換が容易になる。

他の実施態様では，既存のプログラムの命令文または命令のうち，定義体に対する物理的な入出力処理を行う命令文または命令を，上記複数のプログラム構成要素のうちパレットを統括的に制御するためのプログ

ラム構成要素（バレット連鎖関数プログラム；メイン・プログラム）の一部とする。

好ましくは、既存のプログラムから、一の基本構造または複数の基本構造の組合せによって上記既存のプログラムの処理の流れを表す処理経路図を作成する。処理経路図を作成しておくことによって、既存のプログラムの各命令文および単語のそれぞれについて、Lyeの開発手法に従うプログラムにおける位置づけを容易に把握することができる。

たとえば、上記処理経路図に、2以上の異なる基本構造につながる経路または終了処理につながる経路が表されている場合には、その経路の分岐点における分岐条件を表す条件命令文または条件命令を上記既存のプログラムから抽出する。抽出した条件命令文または条件命令を、上記分岐点の分岐元におけるバレットの複数のプログラム構成要素のうちの経路判断のためのプログラム構成要素（経路作用要素プログラム）に従う形に変換する。条件命令文または条件命令のLyeの開発手法に従うプログラムにおける位置づけの一つが把握される。

既存のプログラムに用いられる条件命令文または条件命令には、上述のLyeの方法論に従うプログラムにおいて経路の分岐条件として位置づけられるもののほか、表示画面にデータ出力する場合の条件（出力条件

）およびファイル等からデータを読み込む場合の条件（入力条件）として位置づけられるものや，所定の条件によって処理結果を異ならせる場合の条件として位置づけられるものもある。

既存のプログラムの記述に基づいて，データ入力条件またはデータ出力条件を表す条件命令文または条件命令を検索し，データ入力条件またはデータ出力条件を表す条件命令文または命令が抽出された場合には，抽出した条件命令文または条件命令を，そのデータ入力条件またはデータ出力条件が作用するパレットにおける複数のプログラム構成要素のうち入力作用または出力作用のためのプログラム構成要素（入力／出力作用要素プログラム）に従う形に変換する。

既存のプログラムの記述に基づいて，等価単語の判断条件を表す条件命令文または条件命令を検索し，等価単語の判断条件を表す条件命令文または条件命令が抽出された場合には，抽出した条件命令文または条件命令を，その等価単語が割付けられているパレットにおけるプログラム構成要素のうちデータ生成のためのプログラム構成要素（論理要素プログラムの成立条件）に従う形に変換する。

この発明は，既存のプログラムを，その既存のプログラムが達成する機能を維持して，Lyeeの開発手法に従う構造を持つ新たなプログラム（Lyeeプログラム）

に再生するプログラム再生装置も提供する。

この発明によるプログラム再生装置は、Lyeプログラムを構成する複数のプログラム構成要素を記憶する第1の記憶手段、既存のプログラムに基づいて作成される処理経路図から得られるパレット連鎖関数情報、基本構造情報、パレット情報、定義体情報、経路情報およびコマンド情報を記憶する第2の記憶手段、与えられる既存のプログラムを記憶する第3の記憶手段、第3の記憶手段に記憶された既存のプログラムに用いられている単語を、上記第2の記憶手段に記憶されたパレット連鎖関数情報、基本構造情報、パレット情報および定義体情報に基づいて、その単語が関連する基本構造に含まれるW04パレット、W02パレットおよびW03パレットのうち一または複数のパレットに割付ける第1の割付け手段、および第3の記憶手段に記憶された既存のプログラムの各命令文または命令を、上記第2の記憶手段に記憶された経路情報およびコマンド情報に基づいて、その命令文または命令に含まれる単語が割付けられた一または複数のパレットにおける上記第1の記憶手段に記憶された複数のプログラム構成要素のいずれかに従う形に変換する変換手段を備えている。

この発明によると、既存のプログラムにおいて用いられている単語が、既存のプログラムに基づいて作成

される処理経路図から得られる種々の情報（パレット連鎖関数情報，基本構造情報，パレット情報および定義体情報）に基づいて，W 0 4 パレット，W 0 2 パレットおよび W 0 3 パレットのうち一または複数のパレットに割付けられるので，いずれの単語について Lyee の方法論に従うプログラムを作成すべきであるかを容易に把握することができる。既存のプログラム中の命令文または命令は，経路情報およびコマンド情報から，複数のプログラム構成要素のうちのいずれに従う形に変換すべきであるかが把握される。

プログラム再生装置には，既存のプログラムの命令文または命令を読み込むための読み込み手段が備えられる。読み込み手段によって読み込まれた既存のプログラム命令文または命令が，Lyee の方法論に従うプログラム構成要素（テンプレート）中に用いられる。

さらにこの発明は，上述したプログラム再生装置を制御するためのプログラムを記録した媒体，特に既存のプログラムに用いられている単語をパレットに割付ける処理をプログラム再生装置に実行させるプログラム（割付けプログラム），および既存のプログラムの命令文または命令を，Lyee の方法論に基づく複数のプログラム構成要素のいずれかに従う形に変換する処理をプログラム再生装置に実行させるプログラム（変換プログラム）を記録した媒体を提供している。プログ

ラム記録媒体には、磁気ディスク、光ディスク、光磁気ディスク、磁気テープ、半導体メモリ等を含む。

図面の簡単な説明

第1図はLyeeの方法論に従うソフトウェア・モデルの一般形を示すブロック図である。

第2図はオンライン処理の処理経路図の一例を示す。

第3図はバッチ処理の処理経路図の一例を示す。

第4図はKの処理経路図の一例を、オンライン処理T0の処理経路図とともに示す。

第5図は一般的なソフトウェアについての処理経路図の一部を示す。

第6図はLyeeの方法論における単語の取扱い（単語の相互の関係）を示す概念図である。

第7図は基本構造ごとに必要とされるプログラムの種類と、その位置づけを示す。

第8図はW04パレット関数プログラムの処理の流れを示すフローチャートである。

第9図はW02パレット関数プログラムの処理の流れを示すフローチャートである。

第10図はW03パレット関数プログラムの処理の流れを示すフローチャートである。

第11図は位相要素プログラム（テンプレート）の

処理の流れを示すフローチャートである。

第 1 2 図は論理要素プログラム (W 0 4) (テンプレート) の処理の流れを示すフローチャートである。

第 1 3 図は出力作用要素プログラム (テンプレート) の処理の流れを示すフローチャートである。

第 1 4 図は経路作用要素プログラム (テンプレート) の処理の流れを示すフローチャートである。

第 1 5 図は構造作用要素プログラム (テンプレート) の処理の流れを示すフローチャートである。

第 1 6 図は論理要素プログラム (W 0 2) (テンプレート) の処理の流れを示すフローチャートである。

第 1 7 図は論理要素プログラム (W 0 2 G) (テンプレート) の処理の流れを示すフローチャートである。

第 1 8 図は論理要素プログラム (W 0 2) の具体的なプログラム例を示す。

第 1 9 図はプログラム再生処理の全体的な流れを示す。

第 2 0 図はプログラム再生処理装置の構成を示すブロック図である。

第 2 1 図は従来プログラムの具体例を示す。

第 2 2 図は商品受注画面の一例を示す。

第 2 3 図は商品テーブルの一例を示す。

第 2 4 図は第 2 1 図に示す従来プログラムの処理の

流れを示すフローチャートである。

第 2 5 図は第 2 1 図に示す従来プログラムの処理の流れを示すフローチャートである。

第 2 6 図は形式化処理の詳細な処理の流れを示すフローチャートである。

第 2 7 図はコマンド・テーブル（全体）の一例を示す。

第 2 8 図はコマンド・テーブル（プログラム別）の一例を示す。

第 2 9 図は第 2 1 図に示す従来プログラムの処理を表す処理経路図を示す。

第 3 0 図は処理経路図情報テーブルの一例を示す。

第 3 1 図は第 1 次座標決定処理の詳細な流れを示すフローチャートである。

第 3 2 図は第 1 次座標決定処理後のリー・ベルトの一例を示す。

第 3 3 図は規格化処理の詳細な流れを示すフローチャートである。

第 3 4 図は第 2 次座標決定処理の詳細な流れを示すフローチャートである。

第 3 5 図は第 2 次座標決定処理後のリー・ベルトの一例を示す。

第 3 6 図は命令文の展開処理の詳細な流れを示すフローチャートである。

第 3 7 図は命令列表の一例を示す。

第 3 8 図は I F 文の解析処理の詳細な流れを示すフローチャートである。

第 3 9 図は I F 文の解析処理の詳細な流れを示すフローチャートである。

第 4 0 図は I F 文の解析処理の詳細な流れを示すフローチャートである。

第 4 1 図は Lyee ALL 処理の内容を示すフローチャートである。

第 4 2 図はコピー句展開処理の一例を示す。

第 4 3 図はコピー句展開処理の他の例を示す。

第 4 4 図はコピー句展開処理のさらに他の例を示す。

第 4 5 図はサブ・ルーチン展開処理の一例を示す。

第 4 6 図はクリーニング処理の一例を示す。

第 4 7 図は変則修正処理の一例を示す。

第 4 8 図は命令文の一行化処理の一例を示す。

第 4 9 図は命令文の一行化処理の他の例を示す。

第 5 0 図は単階層化処理の一例を示す。

発明を実施するための最良の形態

1. Lyeeの方法論に従うソフトウェア・モデル

人間がコンピュータに要求する情報処理の基本的な役割は、得ようとする情報を、表示装置の表示画面に表示させたり、プリンタから出力されるプリントに印字させたり、記憶装置に記憶させたりすることにある。得ようとする情報（情報が無いことも含む）を、コンピュータはアウトプットする。Lyeeの方法論に従うソフトウェア・モデルでは、その出力部分（出力に関するソフトウェア部分）を出発点とする。ソフトウェアの出力に関する部分を、W04パレットと呼ぶ（パレット：意味を運ぶものという意味）。

ソフトウェアの出力に関する部分によって制御されるコンピュータの動作（表示画面への情報処理結果の表示等）に応じて、次に行うべき情報処理のための新たな情報がコンピュータに入力される。たとえば、ソフトウェアの出力に関する部分（W04パレット）によって出力された情報が、人間が認識可能な出力情報（表示画面に表示される数値等）であれば、人間はその意味を認識し、判断し、行動する。人間の行動はコンピュータに対しては入力という形で具体化される。入力に関するソフトウェア部分をW02パレットと言う。

コンピュータは、入力に回答してあらかじめ定められた処理を実行する。この処理は、上述のW 0 4パレットによって出力されるべき得ようとする情報を生成するための処理である。入力に回答してコンピュータを動作させるソフトウェア部分をW 0 3パレットとする。

コンピュータは、入力に回答して行われる処理の処理結果を出力する。すなわち、W 0 3パレットの次にはW 0 4パレットが順序づけられる。Lyeeの方法論によると、ソフトウェア・モデルは、基本的には、W 0 4パレットから出発し、W 0 2パレット、W 0 3パレットを経て再びW 0 4パレットに至るという循環の構造を持つ。この循環構造モデルの一般形が、第1図に示されている。

第1図に示す循環構造モデル（以下、「基本構造」という）において、W 0 3パレットは、サブ・パレットであるW 0 2 GおよびW 0 4 Pを含む。W 0 2 Gサブ・パレットは、W 0 2パレットからW 0 3パレットに移されるデータを取込む（Get）等の処理を行うソフトウェア部分である。W 0 4 Pサブ・パレットは、所定のデータ処理によって新たなデータを作成するソフトウェア部分である。Lyeeのソフトウェア・モデルでは、基本的に、W 0 2パレットによって受け付けられた情報がW 0 3パレットのW 0 2 Gサブ・パレットに

取込まれる。W 0 4 P サブ・パレットにおいてあらかじめ定められた処理（四則計算等）が行われる。W 0 4 P サブ・パレットの情報がW 0 4 パレットに移され、W 0 4 パレットによって出力される（表示画面に表示される、ファイル、データベース等に記録される）。

Lyee方法論に従うソフトウェア・モデルにおいて、W 0 4 パレット、W 0 2 パレットおよびW 0 3 パレットからなる基本構造は、W 0 4 パレットによって出力されるデータの種類によって、次の3つに分類される。

その一は、W 0 4 パレットによって出力されるデータが人間に関与するもの（たとえば、表示装置の表示画面に出力される）である場合である。そのようなW 0 4 パレットを含む基本構造を、包括的にオンライン処理T 0と呼ぶ。オンライン処理T 0におけるW 0 4 パレット、W 0 2 パレットおよびW 0 3 パレットを、それぞれT 0 W 0 4 パレット、T 0 W 0 2 パレット、T 0 W 0 3 パレットと呼ぶ。T 0 W 0 3 パレットは、T 0 W 0 2 G サブ・パレットとT 0 W 0 4 P サブ・パレットとを含む。

その二は、W 0 4 パレットによって出力されるデータが人間に関与しないもの（たとえば、ファイル等の記憶媒体に出力されるデータ等）である場合である。

そのような W 0 4 パレットを含む基本構造を，包括的にバッチ処理 T 1 と呼ぶ。バッチ処理 T 1 における W 0 4 パレット，W 0 2 パレットおよび W 0 3 パレットを，それぞれ T 1 W 0 4 パレット，T 1 W 0 2 パレット，T 1 W 0 3 パレットと呼ぶ。T 1 W 0 3 パレットは，T 1 W 0 2 G サブ・パレットと T 1 W 0 4 P サブ・パレットとを含む。

その三は，W 0 4 パレットによって出力されるデータが，最終的に得ようとするデータでない場合（たとえば，T 0 W 0 4 パレットによって表示画面に出力するデータを作成するために，その前処理としてあらかじめ所定のデータを作成しておくような場合）である。そのような W 0 4 パレットを含む基本構造を，包括的に K（Knowledge）と呼ぶ。一般に，Lyee の方法論に従うソフトウェアを最初から作成する場合には，この K の基本構造は現れない（現れないようにソフトウェアを作成することができる）。既存（従来）のプログラムを Lyee の方法論に従うプログラムに再構築する場合に，従来のプログラム構造を Lyee のプログラム構造に当てはめるために K の基本構造が必要とされる。K の基本構造は，従来プログラムのプログラマが作り出した単語を取扱うソフトウェア部分，またはエラー処理を行うソフトウェア部分として位置づけられる。

K における W 0 4 パレット，W 0 2 パレットおよび

W 0 3 パレットを，それぞれ K - W 0 4 パレット，K - W 0 2 パレット，K - W 0 3 パレットと呼ぶ。K - W 0 3 パレットは，K - W 0 2 G サブ・パレットと K - W 0 4 P サブ・パレットを含む。

W 0 4 パレット，W 0 2 パレットおよび W 0 3 パレットからなる基本構造を用いて，所定の処理を行うソフトウェアを視覚的に表現することができる。基本構造によってソフトウェアの処理の流れを表すものを，処理経路図と呼ぶ。オンライン処理 T 0 を含む処理経路図の一例が第 2 図に示されている。この処理経路図は，T 0 W 0 4 パレット 11，T 0 W 0 2 パレット 12 および T 0 W 0 3 パレット 13 から構成される基本構造（循環構造）を含んでいる。

T 0 W 0 4 パレット 11 と T 0 W 0 2 パレット 12 との間には，表示装置の表示画面を表すブロック 14 がある。T 0 W 0 4 パレット 11 によって表示画面 14 上に文字，数字等が表示される。

表示画面 14 に表示される文字，数字等を見て人間が入力装置を用いて，文字，数字等を入力する。入力された文字，数字等が T 0 W 0 2 パレット 12 によって取扱われる。

T 0 W 0 2 パレット 12 に入力する文字，数字等が用いられて T 0 W 0 3 パレット 13 において所定の処理が行われる。すなわち，T 0 W 0 3 パレット 13 の T 0 W

0 2 G サブ・パレット（後述する位相要素プログラム）が，文字，数字等を表すデータを T 0 W 0 2 パレット 12 から受取る。T 0 W 0 4 P サブ・パレットにおいて，T 0 W 0 2 パレットのデータ等が用いられて所定の処理（四則演算等）が行われる。所定の処理が行われたデータ（処理結果）が T 0 W 0 4 パレット 11 に与えられる。T 0 W 0 4 パレット 11 は，T 0 W 0 4 P サブ・パレットのデータ（処理結果）を受取り，表示画面 14 に表示する。

T 0 W 0 2 パレット 12 からのびる経路が 2 つに分かれている。一方が T 0 W 0 3 パレット 13 につながる経路，他方は終了処理につながる経路である。たとえば，表示画面 14 に「終了」ボタンが表示されているときに（そのような表示画面を表示するようなプログラムを作成するとき）に，このような処理経路図が描かれる。コンピュータの入力装置上のエンター・キーが押下されると，T 0 W 0 3 パレット 13 に向かう経路が選択される。「終了」ボタンが押下されると，終了処理に向かう経路が選択される。

バッチ処理 T 1 を含む処理経路図の一例を第 3 図に示す。この処理経路図は，T 1 W 0 4 パレット 21，T 1 W 0 2 パレット 22 および T 1 W 0 3 パレット 23 から構成される基本構造（循環構造）を含んでいる。

T 1 W 0 2 パレット 22 はファイル B 25 に記憶されて

いるデータを読み込む。読み込んだデータが T 1 W 0 3 パレット 23 に移される (T 1 W 0 2 パレット 22 から T 1 W 0 3 パレット 23 の T 1 W 0 2 G サブ・パレットへのデータ移動)。T 1 W 0 3 パレット 23 において所定の処理が行われる (T 1 W 0 4 P サブ・パレットによる T 1 W 0 2 G サブ・パレットのデータの取得および T 1 W 0 4 P サブ・パレットにおける所定の処理)。T 1 W 0 4 パレット 21 がその処理結果を取得する (T 1 W 0 4 P サブ・パレットから T 1 W 0 4 パレット 21 へのデータ移動)。T 1 W 0 4 パレット 21 において、処理結果がファイル A 24 に記録される。

T 1 W 0 2 パレット 22 からのびる経路が 2 つに分かれている。一方が T 1 W 0 3 パレット 23 につながる経路、他方は終了処理につながる経路である。たとえば、ファイル B 25 に含まれる複数のレコードについて、順次 T 1 W 0 3 パレット 23 において所定の処理が行われ、すべてのレコードの読出しおよび所定の処理が終了したときに終了処理に向かう経路が選択されるような場合に、このような処理経路図が描かれる。

K を含む処理経路図の一例を第 4 図に示す。上述のように、K の基本構造 (循環構造) は、W 0 4 パレットによって出力されるデータが、最終的に得ようとするデータでない場合の基本構造である。最終的に得ようとするデータの出力 (T 0 W 0 4 パレットまたは T

1 W 0 4 パレットによる処理による出力) に結付けることが必要であるから、Kを含む処理経路図には必ずオンライン処理 T 0 の処理経路図またはバッチ処理 T 1 の処理経路図の少なくともいずれか一方が描かれる。第 4 図に示す処理経路図には、K の処理経路図とともに、オンライン処理 T 0 の処理経路図 (第 2 図と同じ) が描かれている。K の処理経路図の部分およびオンライン処理 T 0 の処理経路図の部分を、それぞれ破線で囲って示す。

K の処理経路図は、K - W 0 4 パレット 31、K - W 0 2 パレット 32 および K - W 0 3 パレット 33 から構成される基本構造 (循環構造) を含んでいる。

T 0 W 0 3 パレット 13 (T 0 W 0 3 パレット 13 の T 0 W 0 4 P サブ・パレット) からのびる経路が、分岐点 A において T 0 W 0 4 パレット 11 に戻る経路と K - W 0 4 パレット 31 につながる経路とに分岐している。T 0 W 0 3 パレット 13 の T 0 W 0 4 P サブ・パレットにおける判断 (経路の判断) によって、T 0 W 0 4 パレット 11 に戻るべきか、K - W 0 4 パレット 31 に進むべきかが決められる。

同じように、K - W 0 3 パレット 33 (K - W 0 3 パレット 33 の K - W 0 4 P サブ・パレット) からの経路が、分岐点 B において、K - W 0 4 パレット 31 に戻る経路と、T 0 W 0 4 P サブ・パレット (T 0 W 0 3 パ

レット 13 の T 0 W 0 4 P サブ・パレットを意味する) に向かう経路とに分岐している。K - W 0 3 パレット 33 の K - W 0 4 P サブ・パレットにおける判断により, K - W 0 4 パレット 31 に向かうべきか, T 0 W 0 4 P サブ・パレットに向かうべきかが決められる。T 0 と K との間, T 1 と K との間等, 異なる種類の基本構造のつながりを, Lyee の方法論において特にラショナルリティと呼ぶ。

分岐点 A において, K - W 0 4 パレット 31 に進む経路が選択されると, K - W 0 4 パレット 31 から K - W 0 2 パレット 32, K - W 0 3 パレット 33 に進み, ここで処理の処理(四則演算等)が行われることになる。

第 4 図ではオンライン処理 T 0 の基本構造と K の基本構造とのつながり(連鎖)を示したが, バッチ処理 T 1 の基本構造と K の基本構造とのつながりも, 同じようにして, 処理経路図を描くことができる。

第 5 図は, 所定の処理を行うソフトウェアについての一般的な処理経路図の一部を示すものである。コンピュータに所定の処理を行わせるソフトウェアについて処理経路図を描くと, 一般的には, 第 5 図に示すように, 多数の基本構造(オンライン処理 T 0 の基本構造, バッチ処理 T 1 の基本構造, K の基本構造)を含む処理経路図が描かれる。

Lyee のソフトウェア・モデルによって処理される情

報の単位は、画面に表示される項目や、ファイル中に現れるデータ項目である。これらは、コンピュータの処理においてはデータ・フィールドとして位置づけられる。Lyeの方法論において、データ・フィールドに何らかの値がセットされた状態を有意状態（有意性）と呼ぶ。データ・フィールドに何らかの値をセットする処理を、成立処理と呼ぶ。何らかのデータがセットされるデータ・フィールド（有意性があるかどうかを判断することができる単位、ソフトウェアが扱うデータを管理するための最も基本的な最小単位）または設けられるべきデータ・フィールドを規定したプログラム記述を、単語と呼ぶ。通常、単語（データ・フィールド）は、画面、帳票、ファイル、データベース、電文等に属している（画面の単語、帳票の単語、ファイルの単語等）。単語が属する画面、帳票、ファイル、データベース、電文等を、包括的に「定義体」と名付ける。一般に、定義体には複数の単語が属する。また定義体にはデータ入力に用いられる定義体（たとえば、コンピュータにデータ・ロードすべきデータが格納されているデータベース）と、データ出力に用いられる定義体（たとえば、処理結果を表示する画面）とに大別される。データ入力に用いられる定義体を入力定義体と、データ出力に用いられる定義体を出力定義体と呼ぶ。データの入出力のいずれにも用いられる定義

体を入出力定義体と呼ぶ。入力定義体に属する単語を入力単語と，出力定義体に属する単語を出力単語と，入出力定義体に属する単語を入出力単語と呼ぶ。

Lyeeのソフトウェア・モデルでは，上述の単語を基礎として，単語ごとのプログラムが作成される。さらに詳しくは，Lyeeのソフトウェア・モデルにおいて扱われるすべての単語について，上述のT0，T1またはKにおける，W04パレット，W02パレットまたはW03パレットのいずれかのパレットによって扱われるべき単語であるかが，あらかじめ決められる（単語のパレットへの配置，座標の決定）。

一の単語について有意性があることが判定されると，その単語に格納されたデータを利用した別の単語についての有意性があることの判断が可能になる。すなわち，単語ごとのプログラムを作成することによって，最終的に得ようとする情報が格納される単語の有意性が満たされているかどうかの判定（データ・フィールドにデータが格納されているかどうか），最終的に得ようとする情報を得るために必要とされる別の単語の有意性が満たされているかどうかの判定，といった単語ごとの判定を行うことができる。このような，Lyeeの方法論における単語の取扱い（単語の相互の関係）の一例が，第6図に概略的に示されている。

最終的に得ようとする情報（出力情報A）を得るた

めに（出力情報 A を表すデータが格納されるべきデータ・フィールド a にデータを格納するために），情報 B および情報 C が必要であるとする。情報 B を得るためには情報 D，情報 E および情報 F を必要とし，情報 C を得るためには情報 G および情報 H が必要であるとする。このような関係が複数の単語（データ・フィールド）において存在すれば，各単語において，有意性があるかどうかを判定することにより，最終的に必要とされる出力情報 A が得られるかどうかを判定することができる。Lyee のソフトウェア・モデルでは，このように，単語ごとのプログラムが用意されることになる。

たとえば，出力情報 A は情報 B と情報 C との足し合わせの結果であるとする（ $A = B + C$ ）。すなわち，情報 B と情報 C が得られると，その得られた結果（情報 B および情報 C）を利用して，出力情報 A を得ることができる。 $A = B + C$ が定められている場合，情報 A が格納されるべきデータ・フィールド a（単語 a）を，Lyee の方法論において特に「端点単語」と呼ぶ。単語 a にデータ（情報 A）を格納する（情報 A を生成する）ために用いられる情報（ここでは情報 B および情報 C）が格納されるデータ・フィールド b およびデータ・フィールド c（単語 b および単語 c）を，特に「始点単語」と呼ぶ。端点単語に格納されるべきデー

タと始点単語に単語に格納されるべきデータは異なる場合もあるし（たとえば，上述の場合），同じ場合もある（データ移動の場合）。

上述したように，Lyeeの方法論によると，W04パレット，W02パレットおよびW03パレットから構成される基本構造（循環構造）が，Lyeeプログラムの基本的なソフトウェア構造になる。このソフトウェア構造を実現するための種々のプログラムが用意されることになる。Lyeeの方法論にしたがって作成されるプログラム（以下，Lyeeプログラムと呼ぶ）の要素となるプログラムの種類を説明する。

第7図はLyeeプログラムにおいて，基本構造ごとに必要とされるプログラムの種類と，その位置づけとを示したものである。基本的に，オンライン処理T0の基本構造，バッチ処理T1の基本構造およびKの基本構造のそれぞれについて，これらのプログラムが用意される。第7図の説明において，T0，T1またはKの区別は省略する。

基本構造（循環構造）は，パレット連鎖関数プログラムによって制御される。すなわち，パレット連鎖関数プログラムは，基本構造を構成するパレット（W04パレット，W02パレット，W03パレット）（後述するパレット関数プログラム）を順番に動作させる，そのようなプログラムである。たとえば，コボル・

プログラムを例にとると，パレット連鎖関数プログラムがメイン・プログラムとして位置づけられ，W 0 4 パレット（関数プログラム），W 0 2 パレット（関数プログラム），W 0 3 パレット（関数プログラム）は，そのサブルーチン・プログラムとして位置づけられる。基本的に，パレット連鎖関数プログラムは基本構造ごとに用意される。パレット連鎖関数プログラムは，他の基本構造のパレット連鎖関数プログラムを所定の条件のもとに起動する処理，物理的な定義体を動作させる処理も行う。

パレット連鎖関数プログラムによって制御される（呼出される）のが，W 0 4 パレット関数プログラム，W 0 2 パレット関数プログラムおよびW 0 3 パレット関数プログラムである。W 0 4 パレット関数プログラム，W 0 2 パレット関数プログラムおよびW 0 3 パレット関数プログラムは，それぞれ次に説明する複数のプログラム（要素プログラム）を制御するプログラムである。コボル・プログラムを例にすると，次に説明する各種の要素プログラムは，PERFORM文やCALL文によってW 0 4 パレット関数プログラム，W 0 2 パレット関数プログラムおよびW 0 3 パレット関数プログラムによって呼出されるプログラムという位置づけになる。

上述のように，Lyeeプログラムにおいて，単語（デ

ータ・フィールド) について，それぞれ W 0 3 パレット関数プログラム（以下，W 0 3 パレットと呼ぶ）によって取扱われるべき単語であるか，W 0 4 パレット関数プログラム（以下，W 0 4 パレットと呼ぶ）によって取扱われるべき単語であるか，W 0 3 パレット関数プログラム（以下，W 0 3 パレットと呼ぶ）の W 0 2 G サブ・パレットまたは W 0 4 P サブ・パレットの部分で取扱われるべき単語であるか（属するか）が，あらかじめ決定される（単語のパレットへの割付け，座標の決定）。以下の説明では，これらの単語を，それぞれ，W 0 4 パレット単語，W 0 2 パレット単語，W 0 2 G サブ・パレット単語，W 0 4 P サブ・パレット単語と呼ぶ。

① W 0 4 パレット関数プログラム（W 0 4 パレット）

位相要素プログラム，論理要素プログラム，出力作用要素プログラム，経路作用要素プログラムおよび構造作用要素プログラムを，この順番に制御するプログラムである。出力定義体の種類（画面 A，画面 B など）ごとに，W 0 4 パレット関数プログラムは用意される。W 0 4 パレット関数プログラム（W 0 4 パレット）の処理の流れを第 8 図に示す。以下，W 0 4 パレットによって制御される個々の要素プログラムの処理について説明する。

位相要素プログラム（W 0 4）：第 11 図に処理の流

れを示す。位相要素プログラム（W 0 4）は，前パレット（W 0 4パレットの前パレットであるからW 0 4 Pサブ・パレット）に属する（割付けられている）単語（W 0 4 Pサブ・パレット単語）のそれぞれについて，空であるかどうか（有意性が成立しているかどうか（データ・フィールドにデータが格納されているかどうか））を判定する。そして，W 0 4 Pサブ・パレット単語について有意性が成立していれば（データ・フィールドにデータが格納されていれば），その単語（データ・フィールド）に格納されているデータを，W 0 4 Pサブ・パレットからW 0 4パレットの対応する単語（データ・フィールド）に位相（データ移動）させる。単語にデータが格納されていない状態を「空」と呼ぶ。W 0 4 Pサブ・パレット単語が「空」であれば，そのまま処理を終了する。

論理要素プログラム（W 0 4）：第12図に処理の流れを示す。W 0 4パレット単語のそれぞれについて，出力編集を行うプログラムである。たとえば，出力すべきデータがコード情報によって単語（データ・フィールド）に格納されている場合に，そのコード情報を言語に変換して表示画面に出力したり，数値を帳票に出力する場合に3桁ごとにアポストロフィ（，）を付して出力する（たとえば，1，000）といった編集処理を行うプログラムである。「編集処理1」，「編

集処理 2」に編集のための処理（編集処理のためのプログラム記述）が位置づけられる。W 0 4 パレット単語が空であるか（データ・フィールドにデータが格納されていない）、空でないか（データ・フィールドにデータが格納されている）によって編集処理を異ならせることができる。「成立？」の判断は、編集処理が実行できたかどうかを判断するものである。編集処理が実行できたならば「成立処理」（たとえば、そのまま処理を続行する）に進み、編集処理が実行できなかったならば、「不成立処理」（たとえば、エラー・フラグを立てる）に進む。

出力作用要素プログラム：第 13 図に処理の流れを示す。画面や帳票やファイルといった出力定義体ごとに用意されるプログラムである。出力定義体にデータを出力するかどうかの判断（単語に格納されたデータを表示画面に出力させるべきかどうか等）を行い、その判断に基づく処理（出力させるべきデータであれば、表示画面に表示させる、出力させるべきでないデータであれば、画面表示を見送る等）を行う。

経路作用要素プログラム（W 0 4）：第 14 図に処理の流れを示す。上述したように、一のパレットによる処理から 2 つ以上の他のパレットのいずれかのパレットの処理に移るように L y e e プログラムが作成されることがある（分岐）。経路作用要素プログラム（W 0 4

) は、どのような条件によっていずれの W 0 2 パレットの処理に進むべきであるかを規定するプログラムである。たとえば、処理経路図において W 0 4 パレットから W 0 2 パレットに進む経路が分岐している場合に、その分岐の数に応じた数の経路作用要素プログラムが用意される。経路作用要素プログラムからパレット連鎖関数プログラムに指示が与えられ、その指示に基づいてパレット連鎖関数プログラムが次に動作させるべき W 0 2 パレットを動作させる。

構造作用要素プログラム：第 15 図に処理の流れを示す。すべての W 0 4 パレット単語について上述の出力作用要素プログラムによって出力処理が行われたことに応答して、その後、W 0 4 パレット単語および W 0 4 G サブ・パレット単語（データ・フィールド）のデータをクリアする処理と、不成立処理が行われた場合に立てられるフラグをオフにする処理とを行うプログラムである。

上述のように Lyee プログラムは、基本的に、単語を単位として作成されるプログラムの集合体である。W 0 4 パレットに属する（割付けられた）単語（W 0 4 パレット単語）について、それぞれ上述の位相要素プログラム、論理要素プログラム（W 0 2）および出力作用要素プログラムが作成される。W 0 4 パレットに割付けられている単語が属する定義体（ここでは、出

力定義体)について,それぞれ構造作用要素プログラムが作成される。経路作用要素プログラムは,上述のように基本構造(循環構造)を構成するW04パレットからのびる経路の分岐の数によってその数が変化する。

② W02パレット関数プログラム(W02パレット)

入力作用要素プログラム,論理要素プログラムおよび経路作用要素プログラムをこの順番に制御する。入力定義体の種類ごとにW02パレット関数プログラムは用意される。第9図にW02パレット関数プログラム(W02パレット)の処理の流れを示す。以下,W02パレットによって制御される個々の要素プログラムの処理について説明する。

入力作用要素プログラム:画面やファイルといった入力定義体ごとに用意されるプログラムである。入力定義体からデータを読み込むかどうかの判断(データベースからデータを読み込むべきかどうか等)を行い,その判断結果に基づく処理(読み込むべきデータをファイルから読出し,W02パレットに割付けられている所定の単語に格納する等)が行われる。第13図に示す出力作用要素プログラムと同様の処理の流れとなる。

論理要素プログラム(W02):第16図に処理の流れを示す。W02パレットに割付けられている単語に格納されたデータの属性(型;タイプ)をチェックす

るプログラムである。たとえば，所定の単語（ユーザが入力装置を用いて入力するデータが格納されるべきデータ・フィールド等）に格納されるべきデータの型は数値である，と定められているのにも関わらず，文字データがユーザによって入力された場合の処理（属性検査）等が，論理要素プログラム（W 0 2）によって行われる。W 0 2 パレット単語と同じ数の論理要素プログラム（W 0 2）が用意される。

経路作用要素プログラム（W 0 2）：W 0 2 パレットから W 0 3 パレットに進む経路が分岐している場合に，どのような条件によって，いずれの W 0 3 パレットの処理に進むべきであるかを規定するプログラムである。分岐の数に応じた数の経路作用要素プログラムが用意される。第 14 図に示す処理と流れとなる。

③ W 0 3 パレット関数プログラム（W 0 3 パレット）

位相要素プログラム，論理要素プログラム，位相型論理要素プログラム，R 型位相要素プログラム，論理要素プログラムおよび経路作用要素プログラムを，基本的には順番に制御する。一つのシステムに一つの W 0 3 パレット関数プログラムが用意される。W 0 3 パレット関数プログラム（W 0 3 パレット）の処理を第 10 図に示す。以下，W 0 3 パレットによって制御される個々の要素プログラムの処理について説明する。

位相要素プログラム（W 0 2 G）：W 0 2 パレット

に割付けられている単語のそれぞれについて、有意性が成立しているかどうか（データ・フィールドにデータが格納されているかどうか）を判定するプログラムである。有意性が成立していれば、その単語（データ・フィールド）に格納されているデータを、W 0 2 パレットから W 0 2 G サブ・パレットの単語（データ・フィールド）に位相（データ移動）させる。W 0 2 パレット単語の数と同じ数の位相要素プログラムが用意される。第11図と同様の処理となる。

論理要素プログラム（W 0 2 G）：第17図に処理の内容を示す。W 0 2 G サブ・パレットに割付けられている単語と同じ数の論理要素プログラムが用意される。この論理要素プログラムは、W 0 2 G サブ・パレットに割付けられている単語に有意性があるかどうか（単語のデータ・フィールドにデータが格納されているかどうか）を判断する。そして、有意性があれば（空でなければ）そのまま処理を終了し、有意性がなければ（空であれば）、有意性を成立させるための所定の処理（初期値の設定など）を試みる。有意性がない単語が有意性を持つようになると（データ・フィールドにデータが格納されると）、処理が終了する。論理要素プログラム（W 0 2 G）は必ずしも必要とはされない。

位相要素型論路要素プログラム：W 0 2 G サブ・パ

レットに属する単語のそれぞれについて、有意性が成立しているかどうかを判定するプログラムである。第17図と同様の処理となる（「自己生成」は、データ移動処理）。

ラショナルリティ型位相要素プログラム：ラショナルリティ型位相要素プログラム（以下、R型位相要素プログラムという）は、異なるパレット連鎖関数プログラムによって制御されるパレット関数プログラムによって扱われる単語（データ・フィールド）に格納されるデータを利用する場合（たとえば、第4図に示す処理経路図において、「T0」と「K」とが連鎖しているような場合が該当する）に、そのような単語に格納されたデータを位相（データ移動）させる処理を行うプログラムである。異なるパレット連鎖関数プログラムによって制御されるW04Pサブ・パレット（他のW04Pサブ・パレット）に割付けられている単語に格納されたデータが、このR型位相要素プログラムによってデータ移動される。第11図と同様の処理となる。

論理要素プログラム（W04P）：W04Pサブ・パレットに属する単語と同じ数の論理要素プログラム（W04P）が用意される。この論理要素プログラムは、W04Pサブ・パレットに割付けられている単語に有意性があるかどうか（単語のデータ・フィールドにデータが格納されているかどうか；空かどうか）を

判断する。そして、有意性があればそのまま処理を終了し、有意性がなければ（データ・フィールドにデータが格納されていなければ）、有意性を成立させるための所定の処理（四則演算やデータ移動処理）を試みる。有意性のなかった単語が有意性を持つようになると（データ・フィールドにデータが格納されると）、処理が終了する。第17図と同様の処理となる。

上述のR型位相要素プログラム、位相要素型論理要素プログラムおよび論理要素プログラム（W04P）の処理の結果は、後述するW03パレット関数プログラムの「再起か？」の判断の対象になる（後述する）。

経路作用要素プログラム（W04P）：どのような条件によっていずれのW04パレットの処理に進むべきであるかを規定するプログラムである。分岐の数に応じた数の経路作用要素プログラムが用意される。第14図と同様の処理となる。

W03パレット関数プログラム（第10図）の「再起か？」の処理は、R型位相要素プログラム中の「位相」が成立するまで、さらに、位相要素型論理要素プログラム中の「自己生成」の部分および論理要素プログラム（W04P）中の「自己生成」の部分において新たにデータが格納されるべき単語にデータが格納されるまで、処理を繰返すことを意味する。W03パレツ

トは，Lyeeの基本構造（第4図等を参照）において，次に進むべき基本構造へのつながりの部分であるので，一般に，何らかのデータが単語（W04Pサブ・パレットに割付けられている単語）に格納されることが必要とされる。単語にデータが格納されていない場合には，ここで処理を繰り返し，単語にデータが格納されたところで次の基本構造における処理に進むことになる。

第18図に，要素プログラムの一つである論理要素プログラム（W02）の具体例を示す。このプログラムは，上述の第16図に示す処理構造を満たすように，プログラミングされている。□（空欄）の部分に，W02パレット単語を表す識別子（単語識別子）が格納される（すなわち，W02パレット単語ごとに論理要素プログラム（W02）が作成される）。複数のW02パレット単語について，同じような処理（属性検査）が行われることになる。単語が異なることによって変化するプログラム記述は，一部のみ（単語に格納されたデータが「数値」であれば次の処理に進む，単語に格納されたデータが「文字」であれば次の処理に進む等の判断を行う部分）である。

上述した複数の要素プログラムおよびパレット関数プログラムが，複数の単語（パレットに割付けられた単語）や定義体，経路について，それぞれ作成される

。パレット連鎖関数プログラムが、基本構造ごとに作成される（同じような処理であれば、複数の基本構造について1つのパレット連鎖関数プログラムを作成するようにしてもよい）。Lyeの法論にしたがって作成されたソフトウェアは、すべて上述の複数のプログラムによって構成される。ソフトウェアの処理が異なる場合であっても、作成されるプログラムの種類およびプログラムの処理構造（上述のフローチャートの処理の流れ）は同じとなる。論理要素プログラム（第17図）における「自己生成」の部分の記述や、出力（入力）作用要素プログラム（第13図）における判断の部分（たとえば、出力作用要素プログラムの「実行条件の判断」の部分等）の記述等が、ソフトウェアの機能にしたがって異なるのみである（第8図から第17図に示すような、プログラムのソフトウェアの機能によって変化しない部分（Lyeプログラムを自動生成するときに繰返し用いられる部分）と、「成立？」や「判断」のようにソフトウェアごとの特有の処理が代入される部分とを持つプログラムを「テンプレート」と呼ぶ）。このように、Lyeの法論に従う開発手法によって作成されたソフトウェアは、その構造が明確に規定されているので解析が容易である。しかも新たな機能を追加する場合であっても、その機能を実現する単語（新たな画面出力項目を追加する場合等）（データ・

フィールド) について，基本構造を満たすように（複数の）プログラム（またはプログラム記述）を追加すればよい。ソフトウェアの修正（アップグレード）も迅速に達成される。

2．従来プログラムのLyeeprogramへの再構築

第19図は，Lyeeprogramの開発手法以外の手法で作成された既存のプログラム（以下，従来プログラムという）を，その従来プログラムが達成する機能を維持して，Lyeeprogramの開発手法に従う新たな構造を持つプログラムに再構築（再生）する処理の全体的な流れを示す流れ図である。

プログラム再生処理（プログラム）は，形式化処理（プログラム）31，規格化処理（プログラム）32およびLyeeprogram ALL処理（プログラム）33の順に実行される。これらの処理の一部をコンピュータによる処理ではなく，手作業で行なうこともできる。プログラム再生処理プログラムはCD-ROM等の記録媒体に記録することができる。CD-ROM等の記録媒体に記録されたプログラム再生処理プログラムがコンピュータ（パーソナル・コンピュータ，ワークステーション等）にインストールされると，そのコンピュータはプログラム再生装置として動作可能になる。第20図にプログラム再生装置のハードウェア構成の一例を示す。

プログラム再生装置 1 は CPU 2 を含む。CPU 2 にはバスを介してデータ入力を受け付けるための入力装置 3 (キーボード, マウス等), データ等を可視的に出力するための表示装置 4 (CRT 表示装置, 印字装置等), フロッピーディスク (FD) に記録されたデータ等を読み取り, かつ FD にデータ等を記録する FD ドライブ 5, CD-ROM に記録されたデータ等を読み取る CD-ROM ドライブ 6 およびオペレーティング・システム, プログラム再生処理プログラム, 上述のテンプレート等が記録された外部記憶装置 7 が接続されている。CPU 2 の内部の (または CPU 2 に接続された) 内部メモリ 8 は, プログラム・エリア, バッファ・エリア等を提供する。

工場, 事業所, 家庭等において用いられるコンピュータ・システムに所定の機能を実現させる場合, その処理のためのプログラム (従来プログラム) は, 複数のプログラムから構成されることが多い。大規模システムでは, そのプログラムの数が数百にのぼることもある。プログラム再生処理では, 基本的に, その一つのプログラムを対象に, 上述の 3 つ処理 (形式化処理, 規格化処理および Lyee ALL 処理) を行う。

① 形式化処理 (ステップ 51)

形式化処理では, 従来プログラムのプログラム記述から, 再生処理に必要な情報を抽出してテーブルにま

とめる，従来プログラムを整形する（構文を明確にする），従来プログラム中に含まれる単語の属性（始点単語か，端点単語か，数値か，プログラマが定義した単語か等）を判断する，処理経路図を作成する，といった従来プログラムのLyeeプログラムへの再構築（再生）を容易にするための処理が行われる。形式化処理の詳細は後述する。

② 規格化処理（ステップ52）

規格化処理では，従来プログラム中に含まれる単語のLyeeプログラムの構造上における位置づけを判断する（単語のパレットへの割付け），Lyeeの方法論（ルール）に従わせるために従来プログラム中の命令文を書直す，といった処理が行われる。また，Lyeeプログラムを作成するためのツール（Lyee ALL）に入力（登録）するための情報の作成も，この規格化処理において行われる。規格化処理の詳細は後述する。

③ Lyee ALL処理（ステップ53）

Lyee ALL処理では，上述の形式化処理および規格化処理を経て得られた，従来プログラムの機能を保持しつつ，Lyeeプログラムの構造を満たすのに必要とされる情報を，Lyeeプログラムの基本的な処理構造を表すプログラム（テンプレート）の所定の位置に代入する処理が行われる。Lyee ALL処理の詳細は後述する。

以下の説明では，コボル（COBOL : Common Business

s Oriented Language) によって作成された従来プログラム (Lyeeの開発手法に従わない手法によって作成されたプログラム) から, Lyeeプログラム (Lyeeの開発手法に従って作成されたプログラム) を生成する処理の流れを具体的に説明する。

第21図はコボルによって作成された従来プログラムの一例を示している。説明の便宜上, プログラム文 (命令文) の左に, 後述する手続部 (PROCEDURE DIVISION) とそれ以外の部とに分けて, 行番号を記す。

はじめに, 第21図に示す従来プログラム (コボル・プログラム) の内容について説明しておく。

コボル・プログラムには, 次の4つの部 (ディビジョン: DIVISION) が含まれている。

(i) 見出し部 (IDENTIFICATION DIVISION) ; 第01行

(ii) 環境部 (ENVIRONMENT DIVISION) ; 第03行

(iii) データ部 (DATA DIVISION) ; 第11行

(iv) 手続部 (PROCEDURE DIVISION) ; 第01行

(i) 見出し部 (IDENTIFICATION DIVISION) には, プログラム名 (プログラム ID) が記述される (第02行; PROGURAMU-ID. AAA1.)。このコボル・プログラムは「AAA1」というプログラム名 (プログラム ID) のプログラムである。

(ii) 環境部 (ENVIRONMENT DIVISION) には, 構成節 (CONFIGURATION SECTION) (第04行) と入出力節 (IN

PUT-OUTPUT SECTION) (第07行) とが含まれている。

構成節 (CONFIGURATION SECTION) には, このコボル・プログラムが処理されるべきコンピュータ名 (翻訳用コンピュータ名および実行用コンピュータ名) が記述される。第21図に示す従来プログラムでは, 翻訳用コンピュータ名が「AS400」であること (第05行: SOURCE-COMPUTER. AS400.) および実行用コンピュータ名が「AS400」であること (第06行: OBJECT-COMPUTER. AS400.) が記述されている。入出力節 (INPUT-OUTPUT SECTION) のファイル管理段落 (FILE-CONTROL) (第08行) に, コンピュータ (翻訳用コンピュータおよび実行用コンピュータ) にデータを与えたり, 処理結果を出力するための装置名 (入出力装置名) が記述される。「GAMEN-F」というファイルを「画面-AAA」という入出力装置にセットすること (第09行: SELECT GAMEN-F ASSIGN TO 画面-AAA.) が記述されている。「ORGANIZATION IS TRANSACTION.」 (第10行) は「GAMEN-F」というファイルがトランザクション処理を行うものであること意味する。

(iii) データ部 (DATA DIVISION) には, ファイル節 (FILE-SECTION) (第12行) と作業場所節 (WORKING-STORAGE SECTION) (第19行) とが含まれている。

ファイル節 (FILE-SECTION) には, ファイルと, ファイルを構成するレコードと, レコードに含まれる項

目との関係が記述される。ファイルには、データベース等にセット（記憶，書込）されるファイル，画面等へ出力（表示）されるファイル，画面等から入力されるファイル，帳票等へ出力（表記）されるファイル等，種々のファイルがある。

第21図に示す従来プログラムでは，「GAMEN-F」というファイルは入出力装置「画面－AAA」にセットされるファイルである（第09行）。すなわち，表示装置の表示画面に表示される項目を扱うファイルである。このようなファイルのレコードを定義体「画面」と呼ぶ。その定義体「画面」の種類を識別する識別子（定義体識別子）が後述する「GAMEN-R」である。

「GAMEN-F」は「GAMEN-R」というレコード（集団項目）を持ち，そこに「SHIN-CD」，「SU」，「TANKA」および「KINGAKU」という項目（基本項目）が含まれていることが，データ部（DATA DIVISION）の第13行～第18行に記述されている。基本項目の記述に続く「PIC X」は，その基本項目によって扱われるべきデータが文字データであることを，「PIC S9」は，扱われるべきデータが符号を伴う数値データであることを，それぞれ意味する。（）内の数値は桁数（文字数）を表す。

上述したように，何らかのデータがセットされるデータ・フィールド（ソフトウェアが扱うデータを管理

するための最も基本的な最小単位)を、Lyeeでは「単語」と呼ぶ。上述のファイル節(FILE-SECTION)の記述から、第21図に示す従来プログラムには、「SHIN-CD」、「SU」、「TANKA」および「KINGAKU」という単語が用いられていることが分かる。これらの単語は定義体「GAMEN-R」に属している単語であることが分かる。

作業場所節(WORKING-STORAGE SECTION)には、データ処理の途中結果等を格納しておく領域(作業領域)が定義される。「SHIN-TBL」というデータベースに、「SHIN-CD」および「TANKA」の2つの領域が設けられ、その他に「END-BTN」および「WK」という2つ領域が設けられている。これらの領域も何らかのデータがセットされるデータ・フィールドであるので、Lyeeにおいて「単語」として位置づけられる。

(iv)手続部(PROCEDURE DIVISION)に、実体的な処理内容が記述される。第21図に示す従来プログラムによって表示装置の表示画面に表示される画面例(商品受注画面)を第22図に、あらかじめ用意されるデータベース(商品テーブル)の例を第23図に、手続部(PROCEDURE DIVISION)の記述に基づく従来プログラムの処理の流れを示すフローチャートを第24図および第25図に、それぞれ示す。

表示画面(この画面の識別子(定義体識別子)が「

GAMEN-R」である) 54には、商品コード入力欄 55、数量入力欄 56、単価表示欄 57、金額表示欄 58および終了ボタン 59が表示されている。これらの入力欄および出力欄に対応するデータ・フィールドが、それぞれ SHIN-CD, SU, TANKA, KINGAKUである。商品データベース(商品テーブル) 60(このデータベースの識別子(定義体識別子)が「SHIN-TBL」である)は、商品コード(SHIN-CD)および単価(TANKA)の項目を含んでいる。

第21図の従来プログラムが実行されると、初期化処理が行われた後(ステップ61)、GAMEN-F(表示画面上に画面表示するためのファイル)が読み込まれる(ステップ62)。表示装置の表示画面に、第21図に示す画面 54が表示される。

表示画面 54上の終了ボタン 59は、従来プログラムの処理を終了させる場合にクリックされる。終了ボタン 59がクリックされると、従来プログラムの処理は終了する(ステップ63でYES, ステップ72)。

入力装置(キーボード等)のエンター・キーが押下されると、表示画面 54の商品コード入力欄 55および数量入力欄 56にデータ入力されているかどうか判断される(ステップ63でNO, ステップ64)。第21図に示す従来プログラムは、商品コード入力欄 55および数量入力欄 56にデータが入力されていることを、一連の処理

の前提とする。このため、商品コード入力欄 55 または数量入力欄 56 の少なくともいずれか一方にデータ入力が行われていない場合、エラー処理が行われる（ステップ 64 で N0，ステップ 73；表示画面 54 の金額表示欄 58 に「99999」が表示される）。

表示画面 54 の商品コード入力欄 55 および数量入力欄 56 にデータ入力されている場合（ステップ 64 で YES），入力された商品コードおよび数量を表すデータがワーク・メモリにセットされる。また，入力された商品コードに基づいて商品テーブル 60 が参照される（ステップ 65）。

入力データ（商品コード，数量）が適切なデータであるかどうか判断される（ステップ 66）。たとえば，上述のように，商品コード入力欄 55 はデータ・フィールド「SHIN-CD」に対応しており（商品コード入力欄 55 に入力されたデータがデータ・フィールド「SHIN-CD」に格納される），「SHIN-CD」に格納されるべきデータは 5 文字の文字データでなければならないことがあらかじめ定められている（データ部（DATA DIVISION）の第 15 行）。商品コード入力欄 55 に数値データが入力されたり，商品テーブル 60 の商品コード欄に存在しないコードが入力されたり，数量入力欄 56 に文字データが入力されたりすると，エラー処理が行われる（ステップ 66 で N0，ステップ 74；表示画面 54 の単価表

示欄 57に「99999」が表示される）。

入力データが適切な場合には（ステップ66でYES），商品コード入力欄55に入力された商品コードに対応する商品の単価（商品テーブル60を参照）と，数量入力欄56に入力された数量との乗算結果が，ワーク・メモリにセットされる（ステップ67）。

セットされた乗算結果が10,000（円）を超えているかどうか判断される（ステップ68）。乗算結果が10,000円を超えていれば（ステップ68でYES），算出金額を，乗算結果（商品単価×数量）に0.8を乗算した結果とする（ステップ69）。表示画面54の単価表示欄57には商品の単価が，金額表示欄58には算出金額が，それぞれ表示される（ステップ71）。乗算結果が10,000円以下であれば（ステップ68でNO），金額表示欄58には，乗算結果に0.9を乗算した結果が表示される（ステップ70，ステップ71）。

以下，第21図に示す従来プログラムを，その機能を維持しつつ，LYEEプログラムに再構築（再生）する処理の流れを具体的に説明する。上述したように，プログラム再生処理は，形式化処理，規格化処理，Lyee ALL処理の順に行われる（第19図）。

第26図は，形式化処理（ステップ51）の処理の内容を詳細に示すフローチャートである。これらの処理の多くはコンピュータにより自動的に行うことができる

。

従来プログラムの記述に基づいて、コマンド・テーブル（全体）が作成される（ステップ81）。第27図は、第21図に示す従来プログラム（コボル・プログラム）の記述に基づいて作成されたコマンド・テーブル（全体）を示すものである。

コマンド・テーブル（全体）は、システムを稼働させるすべてのプログラムを対象として作成される。ここでは、説明の便宜のため、第21図に示す従来プログラムがシステムを稼働させるプログラム（プログラム全体）であるとする。

コマンド・テーブル（全体）は、定義体種別、コマンド、命令種別（プログラム言語）、条件、外部ファイル、入出力区分およびバレット所在の欄を備えている。

「定義体種別」欄には、従来プログラムが規定している（従来プログラムの処理によって用いられる）定義体の種類を表すデータが格納される。定義体とは上述のようにコンピュータが扱う入出力媒体を意味し、画面、データベース、ファイル等が含まれる。従来プログラムの環境部（ENVIRONMENT DIVISION）の入出力節（INPUT-OUTPUT SECTION）の記述およびデータ部（DATA DIVISION）の記述から、従来プログラムが規定している定義体の種類が把握される。もちろん、たと

例えば，一つのシステムにおいて異なる種類の画面が用いられているような場合（そのような記述の従来プログラムの場合）には，「画面 1」，「画面 2」といった画面の種類まで区別することができるデータが記述される。データベース（DB）についても同様である。

「コマンド」欄には，従来プログラムにおいて規定されている入出力装置（定義体）を，従来プログラムの記述中から見つけるために用いられるコマンドが格納される。

「命令種別」欄には，後述する「条件」欄や「入出力装置」欄に記述される入出力装置の検索のための記述が，どのようなプログラム言語についての記述（取り決め）であるかを表す。

「条件」欄には，定義体を取扱う入出力装置を従来プログラム中の記述から見つけるための条件が記述される。「ASSIGN TO 画面」は，従来プログラム中の「ASSIGN TO 画面」という記述の次に，定義体「画面」についての入出力装置名が記述されることを意味する。「FROM」は，「FROM」という記述の次に定義体「DB」についての入出力装置名が記述されることを意味する（コボル・プログラムの取決めである）。「入出力装置」欄には，コボル・プログラムにおいて ASSIGN 文節にはファイル名が，FROM 文節にはデータベース名

が記述されることが記述されている。

「入出力区分」欄（入／出）には，「定義体種別」の欄に格納されている定義体が，入力に関する定義体であるか出力に関する定義体であるかを示すデータ（「入」または「出」）が格納される。

「パレット所在」欄には，「定義体種別」欄に格納されているデータおよび「コマンド」欄に格納されているデータ（または「入出力区分」欄に格納されているデータ）に基づいて，パレットの種別が格納される。「定義体種別」欄に格納される定義体の種類によって「T 0」，「T 1」または「K」のいずれであるかが決定される。「コマンド」欄に格納されている命令（または「入出力区分」欄に格納されている区分）によって「W 0 4」または「W 0 2」のいずれであるかが，決定される。たとえば，定義体「画面」はオンライン処理 T 0 において取扱われる定義体である。また定義体「画面」は入力（商品コードの入力，数量の入力）と出力（単価の出力，金額の出力）とに用いられている。定義体種別が「画面」であって，入出力区分が「入」であれば「T 0 W 0 2」が，入出力区分が「出」であれば「T 0 W 0 4」が，それぞれ格納される。コマンド・テーブル（全体）は，技術者が従来プログラムの記述に基づいて作成してもよい。

次に従来プログラムの整理が行われる（ステップ 82

）。第21図に示す従来プログラムは，従来プログラムの整理のための処理（ステップ82）を経た（整理を終えた，または整理の必要のない）プログラムであるので，従来プログラムの整理のための処理（ステップ82）についての詳細は後述する。

コマンド・テーブル（プログラム別）が作成される（ステップ83）。コマンド・テーブル（プログラム別）を第28図に示す。

第28図に示すように，コマンド・テーブル（プログラム別）は，プログラムID，コマンド，外部ファイル，定義体，入出力区分およびパレット所在の欄を備えている。

「プログラムID」の欄には，従来プログラムの見出し部（IDENTIFICATION DIVISION）に記述されたプログラム名（プログラムID）が格納される。システムを稼働させるプログラムはこのプログラム名（プログラムID）によって特定することができる。プログラム名（プログラムID）ごとに，コマンド・テーブル（プログラム別）が作成される。

「コマンド」欄には，上述のプログラムIDによって特定される従来プログラムの手続部（PROCEDURE DIVISION）中に現れる，データの入出力に関連する命令が格納される。

「入出力装置」欄には，上述のコマンド・テーブル

(全体)の「条件」欄の記述にしたがって、従来プログラムから検索される入出力装置名を表すデータが記述される。

「定義体」欄には、データ部 (DATA DIVISION) および手続部 (PROCEDURE DIVISION) の記述に基づいて、「コマンド」欄に格納されたコマンドによって処理される定義体 (画面、DB等) の定義体名が格納される。

たとえば、第21図に示す従来プログラムにおいて、手続部 (PROCEDURE DIVISION) の第06行目のコマンド「WRITE」は「GAMEN-R」に対する処理を行う命令であるので、コマンド「WRITE」に対応する「定義体」の欄に「GAMEN-R」が格納される。

「入出力区分」欄および「パレット所在」欄については、上述のコマンド・テーブル (全体) と同じである。

コマンド・テーブル (プログラム別) は、コマンド・テーブル (全体) に基づいてコンピュータに作成させることができる。もちろん、従来プログラムの記述に基づいて、技術者が作成するようにしてもよい。

処理経路図が作成される (ステップ46)。第29図は、第21図に示す従来プログラムの記述に基づいて作成された処理経路図を示すものである。

作成されたコマンド・テーブル (プログラム別) か

ら，従来プログラムには，T 0 と T 1 の基本構造が含まれていることが把握される。また，従来プログラムには，一般にそのプログラムを作成したプログラマが定義した単語（データ・フィールド）が用いられているので，K の基本構造も存在するものとする。さらに，T 0 の基本構造および T 1 の基本構造には，それぞれエラー処理を行う基本構造（K の一態様である）も付随するものとする。T 0，T 1 のそれぞれについて，W 0 4 パレット，W 0 3 パレットおよび W 0 2 パレットから構成される基本構造（循環構造）を描画し，これらの基本構造を結び付ける（T 0 の基本構造の W 0 3 パレット（W 0 4 P サブ・パレット）と T 1 の基本構造の W 0 4 パレットとの間を経路を表す線によって結ぶ）。K の基本構造を描画し，T 0 の基本構造と結びつける。T 0 の基本構造および T 1 の基本構造のそれぞれについてのエラー処理の K の基本構造を描画し，T 0 の基本構造および T 1 の基本構造にそれぞれ結びつける。第 29 図に示すような処理経路図が作成される。

作成した処理経路図から抽出される情報が，テーブル（処理経路図情報テーブル）にまとめられる。第 30 図は，処理経路図情報テーブルを示すものである。処理経路図情報テーブルには，現プログラム ID（現 PGM ID），処理経路図 ID，連鎖関数 ID，パレット ID，経路

要素 ID，次パレット ID，入出力コマンド要素 ID，コマンドおよび定義体の欄が設けられている。

「現プログラム ID（現 PGM ID）」は，従来プログラムごとに付される ID である。複数の従来プログラムを Lye プログラムに再生する場合に，この現プログラム ID によってそのそれぞれが識別される。

「処理経路図 ID」は，基本構造ごとに付される ID である。上述のように，基本的には，基本構造（循環構造）ごとにパレット連鎖関数プログラムが設けられる。基本構造のそれぞれを統括的に制御するパレット連鎖関数プログラムを識別する ID が，連鎖関数 ID の欄に格納される。

「パレット ID」は，基本構造を構成するパレットのそれぞれを識別するための ID である。複数の基本構造に含まれるパレット（W 0 4 パレット，W 0 2 パレットおよび W 0 3 パレット）のそれぞれを識別することができるように，ID が付与される。

「経路要素 ID」は，パレット ID によって識別されるパレットからのびる経路を識別する ID である。たとえば，第 29 図に示す処理経路図において，T 0 W 0 3 パレット（T 0 W 0 4 P サブ・パレット）からのびる経路は，T 0 W 0 4 パレットと，T 1 W 0 4 パレットと，K W 0 4 パレットと，K（Err）W 0 4 パレットと結付いている（第 29 図において符号 A（分岐点

A) で示す)。これらの経路のそれぞれについて I D が付与される (Rute-1, Rute-2, Rute-3 および Rute-4)。

「次パレット I D」は、経路要素 I D によって識別される経路によって、「パレット I D」欄に格納されているパレット I D によって特定されるパレットに結び付いているパレット (次パレット) を識別する I D である。処理経路図に描かれるパレットのそれぞれに上述のパレット I D が付されているので、このパレット I D を用いて、経路要素 I D によって識別される経路に結び付いているパレット (次パレット) が表される。

「入出力コマンド要素 I D」は、パレット I D によって識別されるパレットが、画面、データベース、ファイル等にアクセスする (データを読む、データを書込む等) パレットである場合に、そのパレットがデータを読む (入力する) ための処理を行うパレットであるか、データを書込む (出力する) ための処理を行うパレットであるかを識別する I D である。データを読む (入力する) ための処理を行うパレットには「WRITE-1」が、データを書込む (出力する) ための処理を行うパレットには「READ-1」が、それぞれ格納される。「コマンド」欄に、従来プログラムの手続部 (PROCEDURE DIVISION) において用いられているデータの入

出力のための具体的な命令（コマンド）が格納される。

「定義体」欄には，上述の「コマンド」欄に格納されているコマンドによって取扱われる定義体名が格納される。

第26図に戻って，従来プログラムおよびコマンド・テーブル（プログラム別）に基づいて，パレット連鎖関数プログラムが扱うべき命令文（プログラム文）の特定処理（パレット連鎖関数取扱命令文特定処理；ステップ84）および第1次座標決定処理（ステップ85）が行われる。これらの処理結果が次に示すリー・ベルト（Lyee Belt）に反映される。第31図は第1次座標決定処理の詳細を示すフローチャートを，第32図はリー・ベルトの一例（第21図に示す従来プログラムから作成されるリー・ベルト）を，それぞれ示している。

リー・ベルト（Lyee Belt）（第32図）には，行番号記述欄，命令種別記述欄，連鎖関数記述欄および従来プログラムの手続部（PROCEDURE DIVISION）が記述されたプログラム記述欄を含む（これらの情報が記述されたテーブルが，リーベルト（Lyee Belt）である）。パレット連鎖関数取扱命令文特定処理（ステップ84）の結果が，連鎖関数記述欄に記述される。第1次座標決定処理（ステップ85）の結果が，プログラム記

述欄に記述される。

「命令種別」記述欄には，プログラム記述欄に記述された命令文（プログラム文）について，そこに含まれている所定の命令の種類が記述される。プログラム記述欄に記述された命令文に含まれている命令が，端点単語に対する処理を行う命令（端点命令）であるか，IF文であるか（IF）であるか，またはその他のデータ入出力に関する命令であるか（CMD）の区別が記述される。たとえば，従来プログラムの手続部（PROCEDURE DIVISION）に記述された命令文の第14行目の「MOVE 99999 TO KINGAKU OF GAMEN-R」は，端点単語「KINGAKU（OF GAMEN-R）」に対する処理を行うプログラム文である。第14行目に対応する命令種別記述欄に「端点命令」（端点命令であることを表すデータ）が記述される。

「連鎖関数」記述欄には，パレット連鎖関数取扱命令文特定処理（ステップ84）の処理結果が記述される。この処理は，上述のパレット連鎖関数プログラムが取扱うべき命令文（プログラム文）を特定する処理である。定義体（画面等）に対する物理的な処理を行う命令文に，所定のマークが対応付けられて格納される。マークが付与された命令文は，パレット連鎖関数プログラムにおいて（パレット連鎖関数プログラムのプログラム記述の一部として）用いられるべき命令文で

あることを意味する。

プログラム記述欄の従来プログラムの手続部 (PROCEDURE DIVISION) の記述中に、第1次座標決定処理 (ステップ85) による処理結果が追記される。以下、第31図を参照して、第1次座標決定処理の詳細を説明する。第1次座標決定処理は、基本的にコンピュータによって行われる。もちろん、この処理の一部を技術者に行わせるようにしてもよい。

従来プログラムの手続部 (PROCEDURE DIVISION) に記述されている従来プログラムの命令文の1行目が、プログラム再生装置1のCPU2 (または内部メモリ8, 記憶手段) に読込まれる (ステップ101, 103)。プログラム再生装置1のカウンタがインクリメントされる。

CPU2に読込まれた命令文中に、単語の記述が含まれているかどうか (単語が含まれているかどうか) が判断される (ステップ104)。命令文中に単語が含まれているかどうかは、読込まれた命令文中に含まれる文字列の一部に、従来プログラムの手続部 (DATA DIVISION) 中に記述された単語を表す文字列と同じ文字列が含まれているかどうかで判断することができる。

CPU2に読込まれた命令文中に単語が含まれていない場合には、カウンタがインクリメントされ、次の

行の命令文がCPU 2に読込まれる（ステップ104でNO，ステップ105，ステップ103）。

命令文中に単語の記述が含まれている場合（ステップ104でYES），その単語が定義体（画面，ファイル，DB（データベース）等）に属する単語（定義体単語）であるか，数値（値）であるか，定義体に属さず，かつ数値でもない単語（一般には，従来プログラムを作成したプログラマが作成した単語；これをワークと呼ぶ）であるかが判別される。また，命令文中の単語が定義体単語である場合には，いずれの定義体に属する単語であるかまで判別される（ステップ106）。

命令文中に含まれる単語がどのような単語であるかの判断では，データ部（DATA DIVISION）の記述と，環境部（ENVIRONMENT DIVISION）のファイル節（FILE-CONTROL）の記述とが参照される。たとえば，第14行目の「MOVE 99999 TO KINGAKU OF GAMEN-R」が読込まれると，その命令文中には「99999」と，単語「KINGAKU（OF GAMEN-R）」が含まれていることが分かる。「99999」は数値（値）であることが分かる。また単語「KINGAKU（OF GAMEN-R）」は，「KINGAKU」に続く定義体についての記述（「OF GAMEN-R」）と，データ部（DATA DIVISION）の記述（「FD GAMEN-R」および「01 GAMEN-F」）と，環境部（ENVIRONMENT DIVISION）のファイル節（FILE-CONTROL）の記述（「SELECT GAMEN

-F ASSIGN TO 画面 - AAA」) から, 「GAMEN-R」 という画面定義体に属する単語であることが分かる。「99999」には数値(値)であることを表す識別子(第1次座標)が, 単語「KINGAKU (OF GAMEN-R)」には, 画面定義体に属する単語であることを表す識別子(第1次座標, 「画面」)が, それぞれ付与される。このようにして, 第1次座標がリー・ベルトのプログラム記述欄中に追記される(「MOVE 99999 (値) TO KINGAKU OF GAMEN-F (画面)」)。

定義体単語, 数値, ワーク以外の単語(第18行目の「ステータス」)には, その旨を表す識別子(第1次座標; ここでは「制御BOX」)が付与される。追記される識別子は, プログラム記述欄に記述されているプログラム(従来プログラム)とともに, 内部メモリ8(記憶手段)に一時的に記憶される。

次に, 定義体種別, 数値またはワークのいずれかの識別子が付与された単語について, その単語が端点単語または始点単語であるかどうか判断される(ステップ107)。端点単語または始点単語のいずれでもない場合(ステップ107でNO), 次の命令文についての処理に移る(ステップ105, ステップ103)。

端点単語または始点単語であるかは, 単語の検出処理(ステップ104)において検出された単語が, MOVE文またはCOMPUTE文等の値を代入するための(参照す

る) 単語と, 値が代入される単語とが用いられる命令文中の単語であるかどうかによって判断される。たとえば, 第22行目の「**COMPUTE WK (ワーク) = TANKA OF GAMEN-R (画面) * SU OF GAMEN-R (画面)**」は, **COMPUTE**文であるので, この命令文には端点単語および始点単語が含まれていることが分かる。**COMPUTE**文の場合, 「**= (イコール)**」の前に位置する単語が, データが代入される単語(端点単語)である。「**= (イコール)**」の後ろに位置する単語がデータを代入するための単語(始点単語)である。単語「**WK**」には端点単語である旨を表す識別子が, 単語「**TANKA**」および「**SU**」には始点単語であることを表す識別子が, それぞれ付される(「**COMPUTE WK (端: ワーク) = TANKA OF GAMEN-R (始: 画面) * SU OF GAMEN-R (始: 画面)**」)。MOVE文の場合には, 「**T0**」の前に位置する単語が始点単語であり, 「**T0**」の後ろに位置する単語が端点単語である。内部メモリ8に記憶されているプログラムに識別子(識別子を表すデータ)が追記される(定義体識別子等が追記されたプログラムが更新される)。

リー・ベルトの作成と, 処理経路図情報テーブルの作成を終えると, 次に規格化処理(第19図, ステップ52)に移る。第33図は規格化処理(ステップ52)の内容の詳細を示すフローチャートである。

規格化処理では，第2次座標決定処理（ステップ91）により，上述のリー・ベルト（第32図）に新たな情報が追記される。さらに，命令の展開処理（ステップ92）およびIF文の解析処理（ステップ92）が行われて，命令列表（後述する）が作成される。作成された命令列表と，上述の処理経路図情報テーブル等から，Lyee ALL登録情報作成処理が行われる（ステップ54）。以下，これらの処理の詳細を順次説明する。

第34図は，第2次座標決定処理（ステップ91）の詳細な流れを示すフローチャートである。第35図は，第2次座標決定処理によって得られるリー・ベルト（更新されたリー・ベルト）を示すものである。第2次座標決定処理（第34図）には，上述の第1次座標決定処理（第31図）と同じ処理が含まれるので，第1次座標決定処理中の処理と同じ処理には同一の符号を付し，重複した説明を省略する。

第2次座標決定処理も，第1次座標決定処理と同様に，コンピュータによって処理される。もちろん，技術者が処理結果を検証する，必要に応じて技術者が処理結果を修正する等を行ってもよい。

検出された単語（単語には，上述の識別子（第1次座標）を表すデータが付随している）のそれぞれについて，パレット連鎖関数種別（T0か，T1かまたはKか）と，パレット種別（W04パレット単語か，W

0 2 パレット単語か，W 0 2 G サブ・パレット単語かまたは W 0 4 P サブ・パレット単語か）とが判別され，その旨を表す識別子が単語に付与される（ステップ 111，単語のパレットへの割付け）。

単語のパレット連鎖関数種別の判断は，C P U 2（または内部メモリ 8；記憶手段）に読込まれた命令文に含まれている単語が属する定義体種別から判断することができる。たとえば，定義体「GAMEN-R」は画面定義体である。画面等の定義体に属する単語はパレット連鎖関数 T 0（パレット連鎖関数種別が「T 0」）として位置づけられる。定義体「GAMEN-R」に属する単語（「SHIN-CD」，「SU」，「TANKA」および「KING AKU」）（これらの単語に続く「OF GAMEN-R」の記述から，定義体「GEMEN-R」に属する単語であることが把握される）のパレット連鎖関数種別は「T 0」と判断される。C P U 2 に読込まれた命令文にこれらの単語が含まれている場合には，その単語に付随するように，識別子（T 0 か，T 1 か，K かを表すデータ）が追記される。

単語のパレット種別の判断は，その単語が入力に関する単語であるか，出力に関する単語であるかによって判断される。たとえば，単語「SHIN-CD（OF GAMEN-R）」および「SU（OF GAMEN-R）」は手続部（PROCEDURE DIVISION）の第 13 行の記述から，入力に関する単語

語であること，すなわち W 0 2 パレット単語であることが分かる。結局，単語「SHIN-CD (OF GAMEN-R)」および「SU (OF GAMEN-R)」には，「T O W O 2」という識別子が付与される（プログラム記述欄に追記される）ことになる（第35図；リー・ベルトの更新）。従来プログラムに含まれる単語のそれぞれが，複数のパレットのいずれかに割付けられる。

第2次座標決定処理（ステップ91）を終えると，命令文の展開処理に移る（ステップ52）。第36図は，命令文の展開（基本構造の充足）処理の流れを示すフローチャートである。第1次座標決定処理（第31図）と同じ処理には同一の符号を付し，重複した説明を避ける。命令文の展開処理もまた，基本的にはコンピュータによって行われる。

C P U 2（または内部メモリ8）に読込まれた命令文について，その命令文が，値を代入するための（参照する）単語（始点単語）と値が代入されるべき単語（端点単語）とが用いられている命令文（MOVE文またはCOMPUTE文等）であるか（端点単語および始点単語を含むプログラム文であるか）どうか判断される（ステップ121）。MOVE文やCOMPUTE文等以外のプログラム文である場合には，カウンタがインクリメントされて，次の行の命令文が読み込まれる（ステップ121でN0，ステップ105）。

読込まれた命令文が MOVE 文や COMPUTE 文等である場合（ステップ 121 で YES），そこに含まれる始点単語の第 2 次座標と，端点単語の第 2 次座標が参照される。Lyee の循環構造（基本構造）を満たしているかどうか判定される（ステップ 122）。

上述したように，Lyee の方法論に従って作成されるプログラム（Lyee プログラム）は，W 0 4 パレット，W 0 2 パレットおよび W 0 3 パレットの順に処理が行われるようにプログラミングされる。従来プログラムでは，このような作成手法に従うプログラム作成が行われていないので，循環構造を満たさないプログラム（命令文）が存在する。命令文の展開（循環構造の充足）処理は，循環構造を満たしていない命令文を検出して，そのような命令文を循環構造を満たすようにするための処理である。

たとえば，第 21 行の「MOVE TANKA OF SHIN-TBL（始：T1W02） TO TANKA OF GAMEN-R（端：TOW04）」は，T 1 W 0 2 パレット単語（TANKA(OF SHIN-TBL)）に格納されるデータを，T 0 W 0 4 パレット単語（TANKA(OF GAMEN-R)）に格納する命令文となっている。この命令文は，Lyee の方法論の見地から見ると，T 1 W 0 2 パレットの次に T 0 W 0 4 パレットが動作する命令文となっている。T 0，T 1，K のそれぞれにおいて W 0 4 パレット，W 0 2 パレット，W 0 3 パレット

をこの順番に動作させるという Lyee の方法論に従うプログラム構造に反する記述である。そこで、命令文の展開処理では、このような命令文が CPU 2 に読み込まれた場合には、その命令文が循環構造を満たすことになるように、命令文（および新たな単語）を追加（または命令文を分解、展開）する処理が行われる（ステップ 123）。第 21 行の命令文について命令文の展開処理を行った結果（命令列表）を第 37 図に示す。

第 29 図に示す処理経路図および第 37 図に示す命令列表を参照しつつ、第 21 行目の命令文（「MOVE TANKA OF SHIN-TBL（始：T1W02） TO TANKA OF GAMEN-R（端：T0W04）」）に対する命令文の展開処理を説明する。

第 21 行目の命令文の始点単語「TANKA（OF SHIN-TBL）」が属するパレットは T 1 W 0 2 パレットである。Lyee の方法論における単語のパレットへの割付けの規則の一つに、W 0 2 パレット単語と同じ単語は W 0 2 G サブ・パレットにも割付けられるという規則がある。T 1 W 0 2 G サブ・パレットに新たな単語「TANKA（OF SHIN-TBL）」を割付ける。T 1 W 0 2 パレットと T 1 W 0 2 G サブ・パレットとを結ぶ命令文が追加される（MOVE TANKA OF SHIN-TBL(T1W02) TO TANKA OF SHIN-TBL(T1W02G)）。

Lyee の方法論における単語のパレットへの割付けの

規則の一つに，W 0 2 G サブ・パレット単語に対応する単語は，W 0 4 P サブ・パレットにも割付けられるという規則がある。T 1 W 0 4 P サブ・パレットに単語「TANKA (OF SHIN-TBL)」を割付ける。T 1 W 0 2 G サブ・パレットと T 1 W 0 4 P サブ・パレットとを結ぶ命令文が追加される (MOVE TANKA OF SHIN-TBL(T 1 W 0 2 G) TO TANKA OF SHIN-TBL(T 1 W 0 4 P境界))。このような場合に割付けられる単語を特に境界単語と呼ぶ。

第 21 行の命令文の端点単語 (TANKA OF GAMEN-F(TOW 0 4)) が，オンライン処理 T 0 の基本構造の単語であるので，バッチ処理 T 1 の基本構造からオンライン処理 T 0 の基本構造にデータを移動 (位相) させる必要がある。この場合，オンライン処理 T 0 の基本構造の W 0 4 P サブ・パレットに，位相先の新たな単語を設ける (TANKA OF GAMEN-R (TOW 0 4 P境界))。このような場合に割付けられる単語を境界単語と呼ぶ。バッチ処理 T 1 の W 0 4 P サブ・パレットとオンライン処理 T 0 の W 0 4 P サブ・パレットとを結びつける命令文が追加される (MOVE TANKA OF SHIN-TBL(T 1 W 0 4 P境界) TO TANKA OF GAMEN-R(TOW 0 4 P境界))。

境界単語に格納されるデータを，W 0 4 P サブ・パレットに格納する必要がある。オンライン処理 T 0 の W 0 4 P サブ・パレットに新たな単語 (TANKA OF GAM

EN-R(TOW04P)) を設ける。境境単語と新たに生成された W O 4 P サブ・パレットの単語とを結ぶ命令文を設ける (MOVE TANKA OF GAMEN-R(TOW04P境境) TO TANKA OF GAMEN-R(TOW04P)) 。

最後に， T O W O 4 P サブ・パレットに新たに割り付けられた単語「TANKA OF GAME-R(TOW04P)」と， T O W O 4 パレット単語「TANKA OF GAMEN-R(TOW04)」とを結ぶ命令文を追加する (MOVE TANKA OF GAMEN-R(TOW04P) TO TANKA OF GAMEN-R(TOW04)) 。

命令列表 (第 37 図) には，命令文の展開処理が行われた行番号，展開された命令文の L y e e プログラムにおける位置づけ (種別) ，命令文が扱われるパレット (パレット関数プログラム) (所在) および展開後の命令文が記述される。命令列表の内容はプログラム再生装置 1 のメモリ (記憶手段) に記憶されるのはいうまでもない。もちろん，命令列表 (命令文の展開処理の処理結果) を，リー・ベルトのプログラム記述欄に記述してもよい (この場合には，リー・ベルトのプログラム記述欄がさらに更新される) 。

次に I F 文の解析処理が行われる (ステップ 93) 。第 38 図 ~ 第 40 図は，I F 文の解析処理の詳細を示すフローチャートである。I F 文の解析処理は，処理経路図情報テーブル (または処理経路図) を利用する。処理経路図情報テーブルに格納された情報を利用してコ

ンピュータに解析処理を行わせることができる。技術者が、処理経路図情報テーブル（処理経路図）に基づいて、コンピュータに情報入力作業を行うようにしてもよい。

I F 文の解析処理は、従来プログラム中に用いられている I F 文の命令文と、L y e e プログラム上における位置づけを判定する処理である。従来プログラム中に現れる I F 文は、①一の基本構造から他の基本構造（または終了処理）に向かう経路を規定している I F 文、②入力条件または出力条件を規定している I F 文、③等価単語（2以上の命令文において同じ端点単語が用いられている場合、その端点単語を特に等価単語と呼ぶ）を制御する I F 文とに大別される。I F 文の解析処理（第38図～第40図）によって、従来プログラム中の I F 文が、L y e e プログラムのいずれの要素プログラムで用いられるべきであるかが把握される。

処理経路図情報テーブル（第30図）の「経路要素 I D」欄（または処理経路図（第29図））から、経路作用要素プログラムによって分岐されるべき分岐点の一つが選択される（ステップ131）。

選択された分岐点がエラー処理の基本構造に向かう分岐点である場合には（ステップ132でYES）、リール・ベルトのプログラム記述欄から、選択された分岐の次の（後ろの）処理であるエラー処理の基本構造の処

理に関連する I F 文が抽出される。すなわち，ここで抽出される I F 文は，所定の条件によってエラー処理を行う処理が選択されるような命令文である。

抽出された I F 文が，上述の選択された分岐点の元となるパレット（たとえば，第 29 図の処理経路図の分岐点 A の元となるパレットは，T O W O 4 P サブ・パレットである）の経路作用要素プログラム（第 7 図，第 14 図参照）中に用いられる実行条件として位置づけられる。抽出された I F 文と，その I F 文が用いられるべきパレットとの関係が，内部メモリ 8 に一時的に記憶される（ステップ 133）。

第 29 図に示す処理経路図において，分岐点 A（分岐点 A では経路が 4 つに分かれている）は，エラー処理に向かう分岐点を含む。エラー処理の基本構造を制御する I F 文として，第 13 行および第 18 行（第 35 図参照）が抽出される（「99999」がエラーを表すものとして把握されるので，「99999」が関連する命令文を制御する I F 文が，エラー処理の基本構造を制御する I F 文として抽出される）。第 13 行の I F 文は，その後の「ELSE」に続く記述から，画面に関連する処理におけるエラー処理への分岐であることが把握される（第 18 行の I F 文は，その後の「ELSE」に続く記述から，D B（データベース）に関連する処理におけるエラー処理への分岐であることが把握される）。第 13 行の I

F 文が，分岐点 A の元である T 0 W 0 4 P サブ・パレットにおける経路作用要素プログラム中に用いられる I F 文として位置づけられる。第 18 行の I F 文は，分岐点 C の元である T 1 W 0 4 P サブ・パレットにおける経路作用要素プログラム中に用いられる I F 文として位置づけられる。

選択された分岐点が，エラー処理の基本構造へ向かう分岐点でない場合（通常の処理を行う T 0，T 1 または K の基本構造への分岐点）には，選択された分岐点の次の処理を行う基本構造の処理に関連する I F 文が従来プログラムの記述中から抽出され，その抽出された I F 文が，選択された分岐点の元のパレットの経路作用要素プログラム中に用いられる命令文として位置づけられる（ステップ 134）。たとえば，第 29 図に示す処理経路図中の分岐点 B は，一方が終了処理の基本構造に向かう分岐点である。T 0 W 0 2 パレットの経路作用要素プログラム（第 14 図）中の「実行条件の判断」の部分に用いられる I F 文として，プログラム記述欄の第 09 行が抽出される。

次に，処理経路図情報テーブル（処理経路図）から，入力作用要素プログラムまたは出力作用要素プログラムによって入力条件または出力条件が規定される可能性のあるパレットのうちの 하나가選択される（ステップ 141）。所定の条件のときにだけデータベースや

画面からデータを読み込むような場合に，入力条件が規定される。所定の条件のときにデータベースや画面にデータを出力するような場合に，出力条件が規定される。

選択されたパレットについて，入力条件または出力条件があるかどうか判断される（ステップ142）。そのような条件がない場合には，次のパレットの選択に進む（ステップ142でNO，ステップ140，ステップ141）。

入力条件または出力条件があれば，そのような入力条件または出力条件を規定しているIF文が，プログラム記述欄の記述から抽出される。ステップ141において選択されたパレットがW02パレットであれば，入力条件を表すIF文が抽出されて選択されたW02パレットの入力作用要素プログラムの「実行条件」として位置づけられる。選択されたパレットがW04パレットであれば，出力条件を表すIF文が抽出されて，選択されたW04パレットの出力作用要素プログラムの「実行条件」として位置づけられる（ステップ143）。第21図に示す従来プログラム（第35図に示すプログラム記述欄のプログラム）には，このようなIF文はない。

最後に，端点単語が等価単語であるようなMOVE文やCOMPUTE文等がプログラム記述欄から検索される（ス

テ ッ プ 151) 。 その よ う な M O V E 文 や C O M P U T E 文 等 が 見
つ か る と , そ の 命 令 文 を 制 御 す る I F 文 が 抽 出 さ れ る
。 M O V E 文 等 に 用 い ら れ て い る 等 価 単 語 (端 点 単 語) に
デ ー タ を 格 納 す る た め に 用 い ら れ る 始 点 単 語 が 属 す る
W O 4 P サ ブ ・ パ レ ッ ト の 論 理 要 素 プ ロ グ ラ ム 中 の 「
成 立 ? 」 に , 抽 出 さ れ た I F 文 が 用 い ら れ る こ と に な
る (ス テ ッ プ 152) 。 す べ て の 等 価 単 語 を 含 む プ ロ グ
ラ ム 文 に つ い て , こ の 処 理 が 行 わ れ る (ス テ ッ プ 153
) 。

た と え ば , 第 35 図 の プ ロ グ ラ ム 記 述 欄 を 参 照 し て ,
第 24 行 お よ び 第 26 行 の い ず れ に も 端 点 単 語 と し て 「 K I
N G A K U O F G A M E N - R (端 : T O W 0 4) 」 が 用 い ら れ て い る
。 端 点 単 語 「 K I N G A K U O F G A M E N - R (端 : T O W 0 4) 」 は
等 価 単 語 で あ る こ と が 分 か る 。 こ の 等 価 単 語 に つ い て
の 命 令 文 (M O V E) を 制 御 す る I F 文 は 第 23 行 の I F 文 で
あ る 。 第 23 行 の I F 文 が , こ の 等 価 単 語 に 対 す る 始 点 単
語 で あ る 「 W K 」 が 属 す る K - W O 4 P サ ブ ・ パ レ ッ ト
の 論 理 要 素 プ ロ グ ラ ム の 「 成 立 ? 」 の 部 分 と し て 位 置
づ け ら れ る 。

上 述 の I F 文 の 解 析 処 理 は , 必 ず し も 上 述 の 順 番 で
行 う 必 要 は な い 。 等 価 単 語 に つ い て の I F 文 の 解 析 処
理 (第 40 図) を は じ め に 行 っ て も よ い し , 入 力 / 出 力
条 件 の I F 文 の 解 析 処 理 (第 39 図) を は じ め に 行 っ て
も よ い 。

命令文の展開処理（ステップ93）によって作成された命令列表が完成し，IF文の解析処理（ステップ94）が終了すると，Lyee ALL登録情報への変換処理に進む（ステップ94）。この変換処理は，上述のIF文の解析処理までの一連の処理によって得られた情報（処理経路図情報テーブルに格納されている情報，従来プログラム中に現れる単語の第2次座標，命令文の展開処理によって設けられた新たな単語の第2次座標，従来プログラム中のIF文のLyeeプログラムにおける位置づけ等）を，後述するLyee ALL処理プログラムによる処理の入力データ形式に変換する処理である。データ形式を（データの並び等）を変更する処理であるので，詳細な説明は省略する。

データ形式の変更（Lyee ALL登録情報の作成）を終えたと，Lyee ALL処理に進む（第19図；ステップ53）。第41図はLyee ALL処理の内容を示すフローチャートである。

上述したように，Lyeeプログラムでは，あらかじめ定められた処理構造を持つパレット連鎖関数プログラム，パレット関数プログラムおよび要素プログラムの一般形（テンプレート）が用意されている。テンプレートの所定の位置（「成立？」や「自己生成」，「判断」等）に，処理経路図情報テーブルの情報，IF文（成立条件情報），従来プログラム中に現れる命令文が

代入される。たとえば，従来プログラム中の MOVE 文や COMPUTE 文が，自己生成情報として，その命令文中の端点単語が属する基本構造における W O 4 P サブ・パレットの論理要素プログラムの「自己生成」に代入される）。従来プログラムが達成する処理機能を持ちつつ，Lyee の方法論に従うプログラム構造をもつ複数のプログラム（第 7 図に示すプログラム）（Lyee プログラム）が完成する（ステップ 96）。

最後に，従来プログラムの整理処理（第 26 図，ステップ 82）を，コボル・プログラムを例にして説明する。

コボルによって作成されるプログラムには，コピー句（コピー文）が含まれていることがある。これは，ライブラリにあらかじめ原始プログラム（原文）の一部を登録しておき，他のプログラム中にそのまま複写して利用するものである。コピー句が用いられている場合，そのコピー句によって指定されている原文を展開する処理（コピー句展開処理）が行われる。

第 42 図は，データ部（DATA DIVISION）におけるコピー句展開処理の様子を，第 43 図および第 44 図は，手続部（PROCEDURE DIVISION）におけるコピー句展開処理の様子を，それぞれ示している。

第 42 図 に示すように，展開前のプログラム記述中に，「COPY A.」および「COPY B.」の記述が含まれ

、コピー句 A (A PROC に続く部分 (枠で囲って示す)) およびコピー句 B (B PROC に続く部分) が別途用意されている場合には、「COPY A.」の部分に A PROC の内容が、「COPY B.」の部分に B PROC の内容が、それぞれ展開される。展開前のプログラム記述 (左側) が、展開後のプログラム記述 (右側) に書き直される。

第 43 図の展開の例についても同様である。「COPY X.」の部分に、X PROC の内容が展開される。第 44 図に示す展開の例についても同様である。「COPY Y.」の部分に、Y PROC の内容が展開される。

従来プログラム中 (メイン・プログラム中) に、サブ・ルーチン (サブ・プログラム) を呼び出す記述が含まれていることがある。メイン・プログラムによってサブ・ルーチンが呼出され、呼出されたサブ・ルーチンの実行が終了すると、再びメイン・プログラムの処理に戻る。この場合、メイン・プログラムには、サブ・ルーチンを呼出す命令 (サブ・ルーチンの CALL) が含まれている。メイン・プログラム中のサブ・ルーチンの CALL に代えて、そのサブ・ルーチンの命令文をメイン・プログラム中に展開 (挿入) する処理 (サブ・ルーチン展開処理) が行われる。第 45 図は、サブ・ルーチン展開処理の様子を示している。

第 45 図 に示すように、展開前のメイン・プログラ

ム中に，サブ・ルーチンを読出すCALL命令（CALL 'SUB001' USING WK-A WK-B WK-C）が含まれ，サブ・ルーチン（プログラムID（PROGRAM-ID.）が「SUB001.」のサブ・ルーチン）が用意されている場合に，メイン・プログラム中のCALL命令に代えて，そのCALL命令によって呼出されるサブ・ルーチンの内容が，メイン・プログラムに展開（挿入）される（第45図に示すサブ・ルーチンにおいて，COMPUTE文に続く部分（「WK-Z=2*WK-X*WK-Y*WK-Y」）がサブ・ルーチンの実質的な部分であるので，この部分のみがメイン・プログラムに展開される）。CALL命令に含まれている「USING WK-A WK-B WK-C」の記述にしたがって，サブ・プログラム中のWK-X，WK-Y，WK-Zが，メイン・プログラムにおいて，それぞれWK-A，WK-B，WK-Cに置き換えられる。

従来プログラム中に不要なプログラム記述（実行され得ないプログラム記述）が記述されている場合には，そのような記述を削除する処理（クリーニング処理）が行われる。従来プログラム中の実行され得ない部分が削除される。

たとえば，第46図に示すように，飛越し命令「GO TO 0」による飛越し先の記述（LABEL-X）が従来プログラム中にあるが，その飛越し先に飛ぶための命令（GO TO LABEL-X）の記述が従来プログラム中に存在しないような場合には，クリーニング処理によってLABEL-X

の記述が削除される。

従来プログラムに用いられる命令には、他の命令に置き換えることができるものがある。たとえば、COBOLにおいてADD、SUBTRACT、MULTIPLYおよびDIVIDEを用いた命令文は、すべてCOMPUTEを用いた命令文に置き換えることができる。SETを用いた命令文は、MOVEを用いて置き換えることができる。最終的に作成するLyeeprogramによって用いる命令をあらかじめ決定しておく場合には、その決定された命令を用いるように、プログラムを修正する（これを変則修正処理と呼ぶ）。

たとえば、最終的に作成するLyeeprogramにおいて「SET」を用いないこと、および「MOVE」を用いることをあらかじめ決定しておいた場合には、第47図に示すように、プログラム中の「SET IDX TO 1」という記述は、「MOVE 1 TO IDX」に置き換えられる。

整形処理が行われる。整形処理では、次の処理が従来プログラムに対して行われる。

(a) 従来プログラム中のピリオドを削除し、スペースを設ける。また、命令文の区切りを明確にするために、命令文の最後にEND文（命令）を付加する。

(b) 複数の単語が含まれている命令文を分解する。たとえば、第48図に示すように、「MOVE A TO B,C」という命令文を、「MOVE A TO B」と「MOVE A TO C

」の2行の命令文に分解する。また、第49図に示すように、IF条件の部分において、IF文の終わりを意味する「END-IF」を追加する。

(c) コメント行を削除する。

(d) 単階層化する。COBOLの場合、PERFORM文等が用いられることによって、プログラムが多階層構造となっていることがある。これを、単階層構造に変換する。たとえば、第50図に示すように、従来プログラム中にPERFORM文が用いられている場合には、PERFORM命令およびそれに対応するEXIT命令を削除し、PERFORM文によって呼出されるプログラム部分を、PERFORM命令の部分に移動させる。

上述したプログラム再生処理では、コボル言語によって作成された従来プログラムをLyeeの開発手法に従うプログラム(Lyeeプログラム)に再構築する方法を一貫して説明したが、他のプログラム言語によって作成された従来プログラムについても、基本的に同様の考え方を適用することができる。すなわち、コボル言語によって作成されるプログラムと同じように、他のプログラム言語によって作成されるプログラム(従来プログラム)にも所定の規則(プログラム記述の約束事)が存在し、その所定の規則を把握することができれば、どのような単語(データ・フィールド)が用いられているのか、その単語はどのような入出力装置(

定義体)に関連する単語であるのか等を把握することができる。パレットに、従来プログラムに現れる単語とLyeの基本構造を満たすための新たな単語を割付け(第1次および第2次座標を決定する)、その単語および命令文を組み込んだテンプレート(テンプレートを用いた種々のプログラム(第7図))を作成することによって、他のプログラム言語を用いて作成された従来プログラムについても、Lyeの方法論に従うLyeプログラムに再構築することができる。

請求の範囲

1. Lyeeの開発手法以外の方法で作成された既存のプログラムを、上記既存のプログラムが達成する機能を維持して、Lyeeの開発手法に従う構造を持つ新たなプログラムに再生する方法であり、

上記既存のプログラムから、そこで用いられている単語を、その単語が関連する定義体を識別した形で抽出し、

抽出した単語を、その単語が関連する定義体の種類に応じて、一または複数のパレットに割付け、

既存のプログラムの各命令文または命令を、その命令文または命令に含まれる単語が割付けられた一または複数のパレットにおけるLyeeの開発手法に従う構造を持つ複数のプログラム構成要素のいずれかに従う形に変換する、

プログラム再生方法。

2. 既存のプログラムの命令文または命令のうち、単語によって表されるフィールドに格納されるべきデータを生成するための命令文または命令を、上記複数のプログラム構成要素のうちデータ生成のためのプログラム構成要素に従う形に変換する、

請求の範囲第1項に記載のプログラム再生方法。

3. 既存のプログラムの命令文または命令のうち、2以上の単語を含む命令文または命令について、

その命令文または命令に用いられている 2 以上の単語のそれぞれが割付けられている 2 以上のパレットの関係が、所定の順序を満たす関係であるかどうかを判定し、

2 以上のパレットの関係が所定の順序を満たしていない場合に、その命令文または命令を複数の命令文または命令に分解し、分解された命令文または命令のそれぞれに新たな単語を含ませ、

分解された命令文または命令のそれぞれに含まれる 2 以上の単語が割付けられる 2 以上のパレットの関係が所定の順序を満たすように、新たな単語を一または複数のパレットに割付ける、

請求の範囲第 1 項に記載のプログラム再生方法。

4. 抽出した単語が端点単語であるか、または始点単語であるかを識別する、

請求の範囲第 1 項に記載のプログラム再生方法。

5. 既存のプログラムの命令文または命令のうち、定義体に対する物理的な入出力処理を行う命令文または命令を、上記複数のプログラム構成要素のうちパレットを統括的に制御するためのプログラム構成要素の一部とする、

請求の範囲第 1 項に記載のプログラム再生方法。

6. 上記パレットは、出力処理に関する W 0 4 パレット、入力処理に関する W 0 2 パレットおよびデータ生

成処理に関する W 0 3 パレットのいずれかであり、

抽出した単語が入力処理に用いられる定義体に関連する場合には、その単語を少なくとも W 0 2 パレットに割付け、

抽出した単語が出力処理に用いられる定義体に関連する場合には、その単語を少なくとも W 0 4 パレットに割付ける、

請求の範囲第 1 項に記載のプログラム再生方法。

7. 既存のプログラムで用いられている単語が関連する定義体の種類に応じて、W 0 4 パレット、W 0 2 パレットおよび W 0 3 パレットの組を設ける、

請求の範囲第 6 項に記載のプログラム再生方法。

8. 上記既存のプログラムで用いられている単語が、定義体に関連していない単語である場合には、その単語を割付けるための W 0 4 パレット、W 0 2 パレットおよび W 0 3 パレットの組を別に設ける、

請求の範囲第 6 項に記載のプログラム再生方法。

9. 上記既存のプログラムにエラー処理の記述が含まれている場合には、そのエラー処理を取扱うための W 0 4 パレット、W 0 2 パレットおよび W 0 3 パレットの組を別に設ける、

請求の範囲第 6 項に記載のプログラム再生方法。

10. Lye の開発手法以外の方法で作成された既存のプログラムを、その既存のプログラムが達成する機能を

維持して，Lyeの開発手法に従う構造を持つ新たなプログラムに再生する方法であり，

既存のプログラムから既存のプログラムによって制御される定義体を抽出し，

抽出した定義体について，W 0 2 パレット，W 0 4 パレットおよびW 0 3 パレットの組から構成される基本構造を動作させるための複数のプログラム構成要素を用意し，

上記既存のプログラムから既存のプログラムで用いられている単語を，その単語が属する定義体およびその単語の入出力区分を識別した形で抽出し，

抽出した単語が属する定義体および入出力区分に応じて，その単語を，その単語が属する定義体についてのW 0 2 パレット，W 0 4 パレットおよびW 0 3 パレットのうちの一または複数のパレットに割付け，

既存のプログラムの単語を含む命令文または命令を，その命令文または命令に単語を通して関連するパレットにおけるLyeの開発手法に従う構造を持つ複数のプログラム構成要素のうちのいずれかに従う形に変換する，

プログラム再生方法。

11. 抽出した単語が入力処理に用いられる定義体に属する場合には，その単語を少なくともW 0 2 パレットに割付け，

抽出した単語が出力処理に用いられる定義体に属する場合には，その単語を少なくともW04パレットに割付ける，

請求の範囲第10項に記載のプログラム再生方法。

12. 抽出した単語が，抽出した定義体のいずれにも属していない単語である場合には，その単語を割付けるための基本構造を動作させる複数のプログラム構成要素を別に用意する，

請求の範囲第10項に記載のプログラム再生方法。

13. 上記既存のプログラムにエラー処理の記述が含まれている場合には，そのエラー処理を取扱うための基本構造を動作させる複数のプログラム構成要素を別に用意する，

請求の範囲第10項に記載のプログラム再生方法。

14. 既存のプログラムの命令文または命令のうち，単語によって表されるフィールドに格納されるべきデータを生成するための命令文または命令を，上記複数のプログラム構成要素のうちデータ生成のためのプログラム構成要素に従う形に変換する，

請求の範囲第10項に記載のプログラム再生方法。

15. 既存のプログラムの命令文または命令のうち，2以上の単語を含む命令文または命令について，

その命令文または命令に用いられている2以上の単語のそれぞれが割付けられている2以上のパレットの

関係が，所定の順序を満たす関係であるかどうかを判定し，

2以上のパレットの関係が所定の順序を満たしていない場合に，その命令文または命令を複数の命令文または命令に分解し，分解された命令文または命令のそれぞれに新たな単語を含ませ，

分解された命令文または命令のそれぞれに含まれる2以上の単語が割付けられている2以上のパレットの関係が所定の順序を満たすように，新たな単語をW04パレット，W02パレットおよびW03パレットのうち一または複数のパレットに割付ける，

請求の範囲第10項に記載のプログラム再生方法。

16. 既存のプログラムの命令文または命令のうち，定義体に対する物理的な入出力処理を行う命令文または命令を，上記複数のプログラム構成要素のうちパレットを統括的に制御するためのプログラム構成要素の一部とする，

請求の範囲第10項に記載のプログラム再生方法。

17. 既存のプログラムから，一の基本構造または複数の基本構造の組合せによって上記既存のプログラムの処理の流れを表す処理経路図を作成する，

請求の範囲第10項に記載のプログラム再生方法。

18. 上記処理経路図に，2以上の異なる基本構造につながる経路または終了処理につながる経路が表されて

いる場合に，その経路の分岐点における分岐条件を表す条件命令文または条件命令を上記既存のプログラムから抽出し，

抽出した条件命令文または条件命令を，上記分岐点の分岐元におけるパレットの複数のプログラム構成要素のうち経路判断のためのプログラム構成要素に従う形に変換する，

請求の範囲第17項に記載のプログラム再生方法。

19. 既存のプログラムの記述に基づいて，データ入力条件またはデータ出力条件を表す条件命令文または条件命令を検索し，

データ入力条件またはデータ出力条件を表す条件命令文または条件命令が抽出された場合に，抽出した条件命令文または条件命令を，そのデータ入力条件またはデータ出力条件が作用するパレットにおける複数のプログラム構成要素のうち入力作用または出力作用のためのプログラム構成要素に従う形に変換する，

請求の範囲第10項に記載のプログラム再生方法。

20. 既存のプログラムの記述に基づいて，等価単語の判断条件を表す条件命令文または条件命令を検索し，

等価単語の判断条件を表す条件命令文または条件命令が抽出された場合に，抽出した条件命令文または条件命令を，その等価単語が割付けられているパレットにおける複数のプログラム構成要素のうちデータ生成

のためのプログラム構成要素に従う形に変換する，

請求の範囲第10項に記載のプログラム再生方法。

21. Lyeeプログラムを構成する複数のプログラム構成要素を記憶する第1の記憶手段，

既存のプログラムに基づいて作成される処理経路図から得られるパレット連鎖関数情報，基本構造情報，パレット情報，定義体情報，経路情報およびコマンド情報を記憶する第2の記憶手段，

与えられる既存のプログラムを記憶する第3の記憶手段，

第3の記憶手段に記憶された既存のプログラムに用いられている単語を，上記第2の記憶手段に記憶されたパレット連鎖関数情報，基本構造情報，パレット情報および定義体情報に基づいて，その単語が関連する基本構造に含まれるW04パレット，W02パレットおよびW03パレットのうち一または複数のパレットに割付ける第1の割付け手段，および

第3の記憶手段に記憶された既存のプログラムの各命令文または命令を，上記第2の記憶手段に記憶された経路情報およびコマンド情報に基づいて，その命令文または命令に含まれる単語が割付けられた一または複数のパレットにおける上記第1の記憶手段に記憶された複数のプログラム構成要素のいずれかに従う形に変換する変換手段，

を備えたプログラム再生装置。

22. 既存のプログラムの命令文または命令を読み込む命令文または命令読み込み手段を備えた，

請求の範囲第21項に記載のプログラム再生装置。

23. 上記第1の割付け手段は，上記読み込み手段によって読み込まれた命令文または命令に用いられている単語が入力処理に用いられる定義体に関連する場合には，その単語を少なくともW02パレットに割付け，

読み込まれた命令文または命令に用いられている単語が出力処理に用いられる定義体に関連する場合には，その単語を少なくともW04パレットに割付ける，

請求の範囲第22項に記載のプログラム再生装置。

24. 上記読み込み手段によって読み込まれた命令文または命令のうち，2以上の単語を含む命令文または命令について，その命令文または命令に用いられている2以上の単語のそれぞれが割付けられている2以上のパレットの関係が，所定の順序を満たす関係であるかどうかを判定する判定手段，

上記判定手段によって2以上のパレットの関係が所定の順序を満たしていないと判定された場合に，読み込まれた命令文または命令を複数の命令文または命令に分解し，分解された命令文または命令のそれぞれに新たな単語を含ませる手段，および

分解された命令文または命令のそれぞれに含まれる

2以上の単語が割付けられる2以上のパレットの関係が所定の順序を満たすように、新たな単語を一または複数のパレットに割付ける第2の割付け手段、

を備えた請求の範囲第22項に記載のプログラム再生装置。

25. 上記読込み手段によって読込まれた命令文または命令に用いられている単語が端点単語であるか、または始点単語であるかを識別する端点単語／始点単語識別手段、

を備えた請求の範囲第22項に記載のプログラム再生装置。

26. 上記第1の記憶手段には、データ生成のためのプログラム構成要素が記憶されており、

上記変換手段は、

既存のプログラムの命令文または命令のうち、単語によって表されるフィールドに格納されるべきデータを生成するための命令文または命令を、上記複数のプログラム構成要素のうち上記データ生成のためのプログラム構成要素に従う形に変換する、

請求の範囲第21項に記載のプログラム再生装置。

27. 上記第1の記憶手段には、W04パレット、W02パレットおよびW03パレットの組を統括的に制御するプログラム構成要素が記憶されており、

上記変換手段は、

既存のプログラムの命令文または命令のうち，画面に対する入出力処理を行う命令文または命令を，上記複数のプログラム構成要素のうち上記W04パレット，W02パレットおよびW03パレットを統括的に制御するためのプログラム構成要素の一部とする，

請求の範囲第21項に記載のプログラム再生装置。

28. 上記第1の記憶手段には，経路判断のためのプログラム構成要素が記憶されており，

上記第2の記憶手段に記憶された経路情報に基づいて，その経路の分岐条件を表す条件命令文または条件命令を上記既存のプログラムから抽出する経路分岐条件命令抽出手段を備え，

上記変換手段は，

抽出された条件命令文または条件命令を，上記経路の分岐元におけるパレットの複数のプログラム構成要素のうち上記経路判断のためのプログラム構成要素に従う形に変換する，

請求の範囲第21項に記載のプログラム再生装置。

29. 上記第1の記憶手段には，入力作用および出力作用のためのプログラム構成要素が記憶されており，

既存のプログラムの記述に基づいて，データ入力条件またはデータ出力条件を表す条件命令文または条件命令を検索するデータ入出力条件命令検索手段を備え，

上記変換手段は、

データ入力条件またはデータ出力条件を表す条件命令文または条件命令が抽出された場合に、抽出された条件命令文または条件命令を、そのデータ入力条件またはデータ出力条件が作用するパレットにおける複数のプログラム構成要素のうち上記入力作用または出力作用のためのプログラム構成要素に従う形に変換する、

請求の範囲第21項に記載のプログラム再生装置。

30. 上記第1の記憶手段には、データ生成のためのプログラム構成要素が記憶されており、

既存のプログラムの記述に基づいて、等価単語の判断条件を表す条件命令文または条件命令を検索する等価単語判断条件命令検索手段を備え、

上記変換手段は、

等価単語の判断条件を表す条件命令文または条件命令が抽出された場合に、その条件命令文または条件命令を、その等価単語が割付けられているパレットにおける複数のプログラム構成要素のうち上記データ生成のためのプログラム構成要素に従う形に変換する、

請求の範囲第21項に記載のプログラム再生装置。

31. Lyeeの開発手法以外の方法で作成された既存のプログラムを、上記既存のプログラムが達成する機能を維持して、Lyeeの開発手法に従うパレット構造を持つ

新たなプログラムに再生するプログラム再生装置を制御するプログラムであって、

上記既存のプログラムの命令文または命令で用いられている単語が属する定義体の種類を、既存のプログラムから得られるパレット連鎖関数情報、基本構造情報、パレット情報および定義体情報に基づいて判別させ、

属する定義体の種類が判別された上記単語について、読込まれた命令文または命令に基づいて、その単語が割付けられるべきパレットの種類を判別させ、

判別された定義体の種類およびパレットの種類に応じて、上記単語を一または複数のパレットに割付けるように上記プログラム再生装置を制御する割付けプログラム、

を記録した記録媒体。

32. 上記単語が用いられている命令文または命令の種類に基づいて、上記命令文または命令で用いられている単語が端点単語であるか、または始点単語であるかを判別させ、

命令文または命令で用いられている単語が、端点単語または始点単語であることが判定された場合に、上記命令文または命令で用いられている単語に端点単語である旨または始点単語である旨を表す識別子を付与するように上記プログラム再生装置を制御する割付け

プログラム，

を記録した請求の範囲第31項に記載の記録媒体。

33. Lyeeプログラムを構成する複数のプログラム構成要素を記憶した記憶装置を備え，Lyeeの開発手法以外の方法で作成された既存のプログラムを，上記既存のプログラムが達成する機能を維持して，Lyeeの開発手法に従うパレット構造を持つ新たなプログラムに再生するプログラム再生装置を制御するプログラムであって，

上記既存のプログラムは，単語が割付けられるべき一または複数のパレットに関する情報が，上記既存のプログラムの命令文または命令で用いられている単語について付与されているものであり，

上記既存のプログラムの命令文または命令を，その命令文または命令に含まれる単語に付与された割付パレットに関する情報によって特定されるパレットにおける複数のプログラム構成要素のいずれかに従う形に変換するように，プログラム再生装置を制御する変換プログラム，

を記録した記録媒体。

34. 上記複数のプログラム構成要素にはデータ生成のためのプログラム構成要素が含まれており，

上記変換プログラムは，

単語によって表されるフィールドに格納されるべき

データを生成するための命令文または命令を，上記複数のプログラム構成要素のうちデータ生成のためのプログラム構成要素に従う形に変換する，

請求の範囲第33項に記載の記録媒体。

35. 上記複数のプログラム構成要素には複数のパレットの組を統括的に制御するプログラム構成要素が含まれており，

上記変換プログラムは，

画面に対する入出力処理を行う命令文または命令を，上記複数のプログラム構成要素のうち複数のパレットの組を統括的に制御するプログラム構成要素の一部とする，

請求の範囲第33項に記載の記録媒体，

36. 上記複数のプログラム構成要素には経路判断のためのプログラム構成要素が含まれており，

上記変換プログラムは，

既存のプログラムから得られる経路情報に基づいて，その経路の分岐条件を表す条件命令文または条件命令を上記既存のプログラムから抽出させ，

抽出された条件命令文または条件命令を，上記分岐元におけるパレットの複数のプログラム構成要素のうち経路判断のためのプログラム構成要素に従う形に変換する，

請求の範囲第33項に記載の記録媒体。

37. 上記複数のプログラム構成要素には入力作用および出力作用のためのプログラム構成要素が含まれており、

上記変換プログラムは、

既存のプログラムからデータ入力条件またはデータ出力条件を表す条件命令文または条件命令を検索し、

データ入力条件またはデータ出力条件を表す条件命令文または条件命令が抽出された場合には、その条件命令文または条件命令を、そのデータ入力条件またはデータ出力条件が作用するパレットにおける複数のプログラム構成要素のうち入力作用または出力作用のためのプログラム構成要素に従う形に変換する、

請求の範囲第33項に記載の記録媒体。

38. 上記複数のプログラム構成要素にはデータ生成のためのプログラム構成要素が含まれており、

上記変換プログラムは、

既存のプログラムから等価単語の判断条件を表す条件命令文または条件命令を検索し、

等価単語の判断条件を表す条件命令文または条件命令が抽出された場合には、その条件命令文または条件命令を、その等価単語が割付けられているパレットにおける複数の複数のプログラム構成要素のうちデータ生成のためのプログラム構成要素に従う形に変換する、

請求の範囲第33項に記載の記録媒体。

39. Lyeeの開発手法以外の方法で作成された既存のプログラムを，上記既存のプログラムが達成する機能を維持して，Lyeeの開発手法に従う構造を持つ新たなプログラムに再生する装置であり，

Lyeeプログラムを構成する複数のプログラム構成要素を記憶する記憶手段，

既存のプログラムを入力する入力手段，

入力された既存のプログラムから定義体を抽出する定義体抽出手段，

入力された既存のプログラムから単語を抽出する単語抽出手段，

入力された既存のプログラムから単語と定義体との関連性を抽出する単語／定義体関連性抽出手段，

入力された既存のプログラムから上記単語抽出手段によって抽出された単語の入出力区分を抽出する入出力区分抽出手段，

上記単語／定義体関連性抽出手段によって抽出された単語と定義体との関連性および上記入出力区分抽出手段によって抽出された単語の入出力区分に基づいて，上記単語抽出手段によって抽出された単語を，その単語が関連する一または複数のパレットに割付ける割付け手段，ならびに

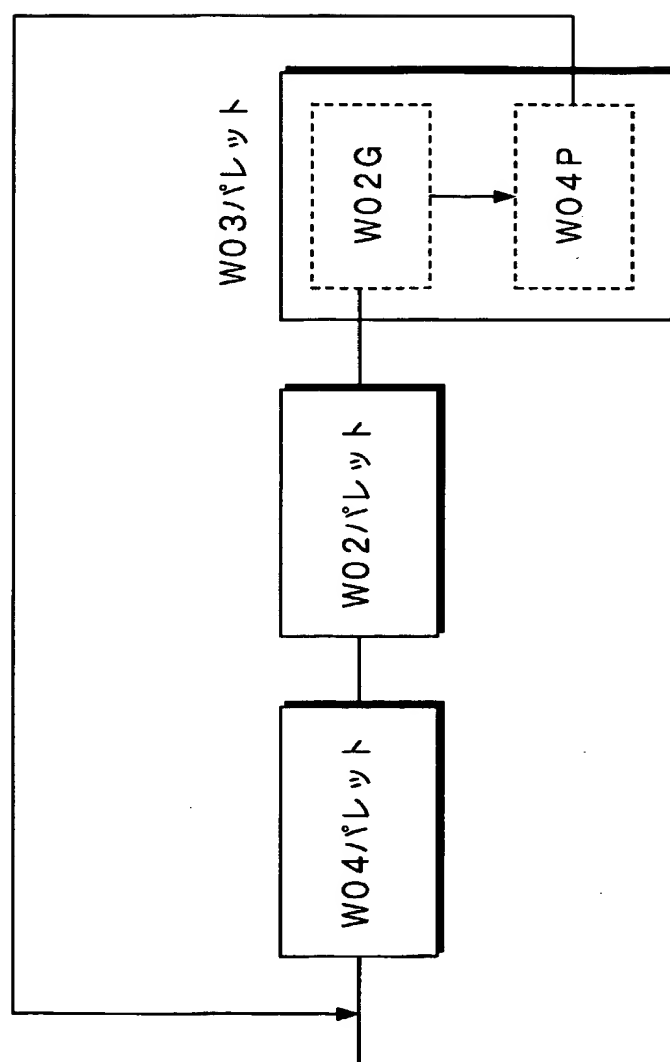
既存のプログラムの各命令文または命令を，その命

令文または命令に含まれる単語が割付けられた一または複数のパレットにおける上記記憶手段に記憶された複数のプログラム構成要素のいずれかに従う形に変換する変換手段、

を備えたプログラム再生装置。

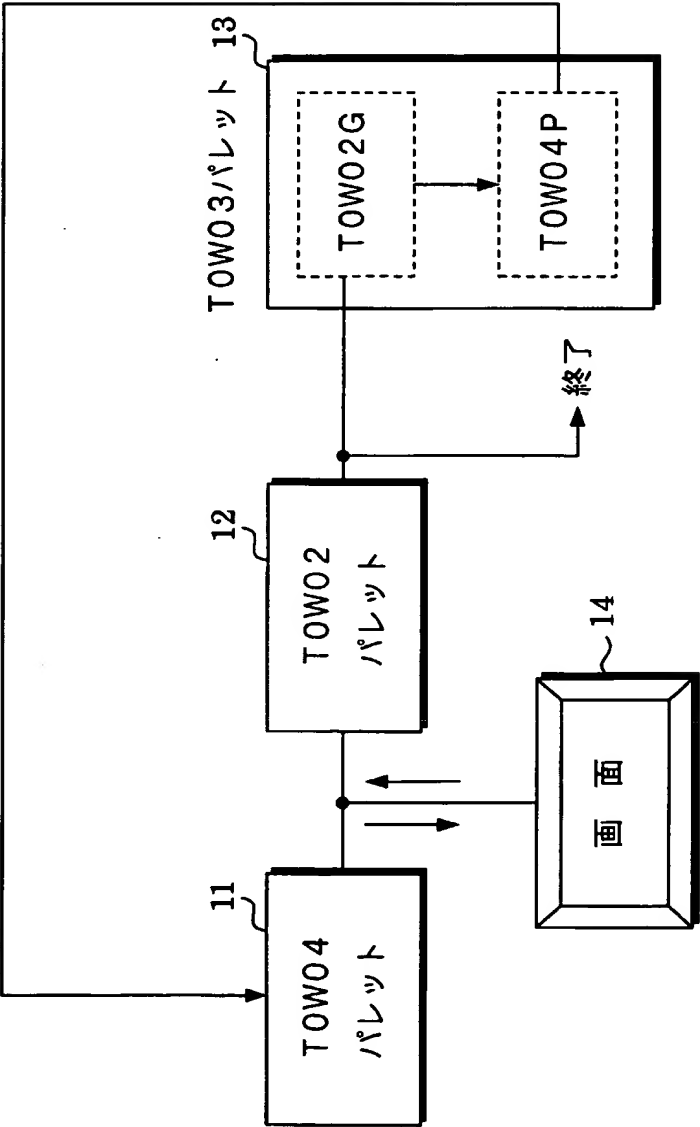
1 / 50

第1図



THIS PAGE BLANK (USPTO)

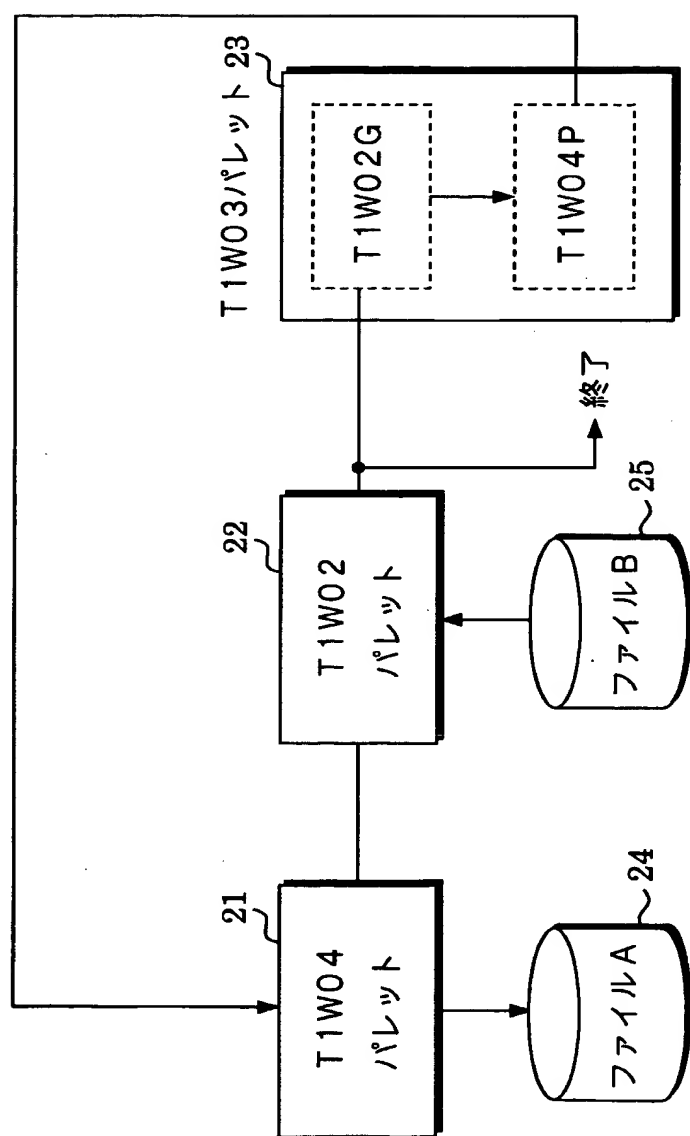
第2図



THIS PAGE BLANK (USPTO)

3 / 5 0

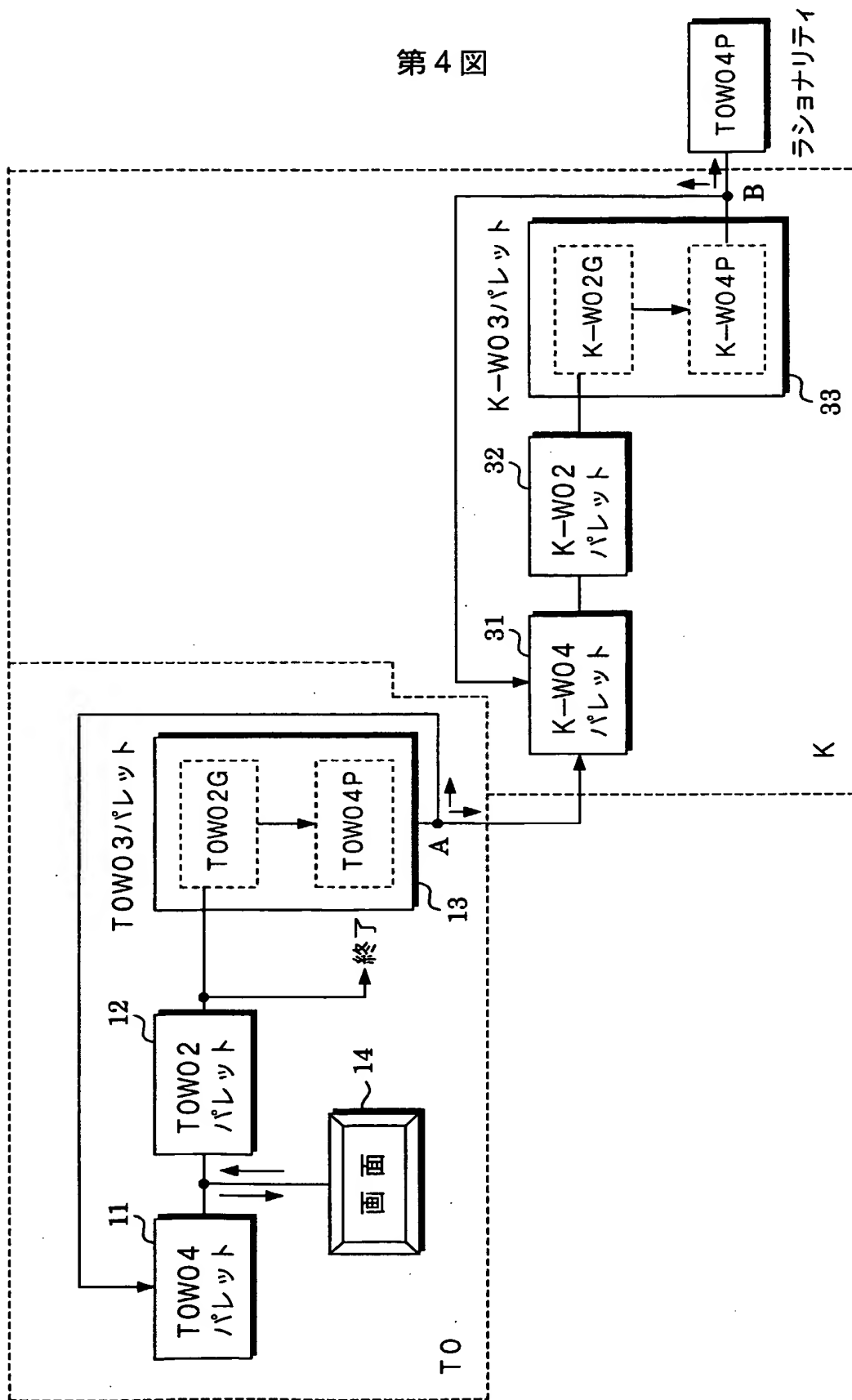
第3図



THIS PAGE BLANK (USPTO)

4 / 50

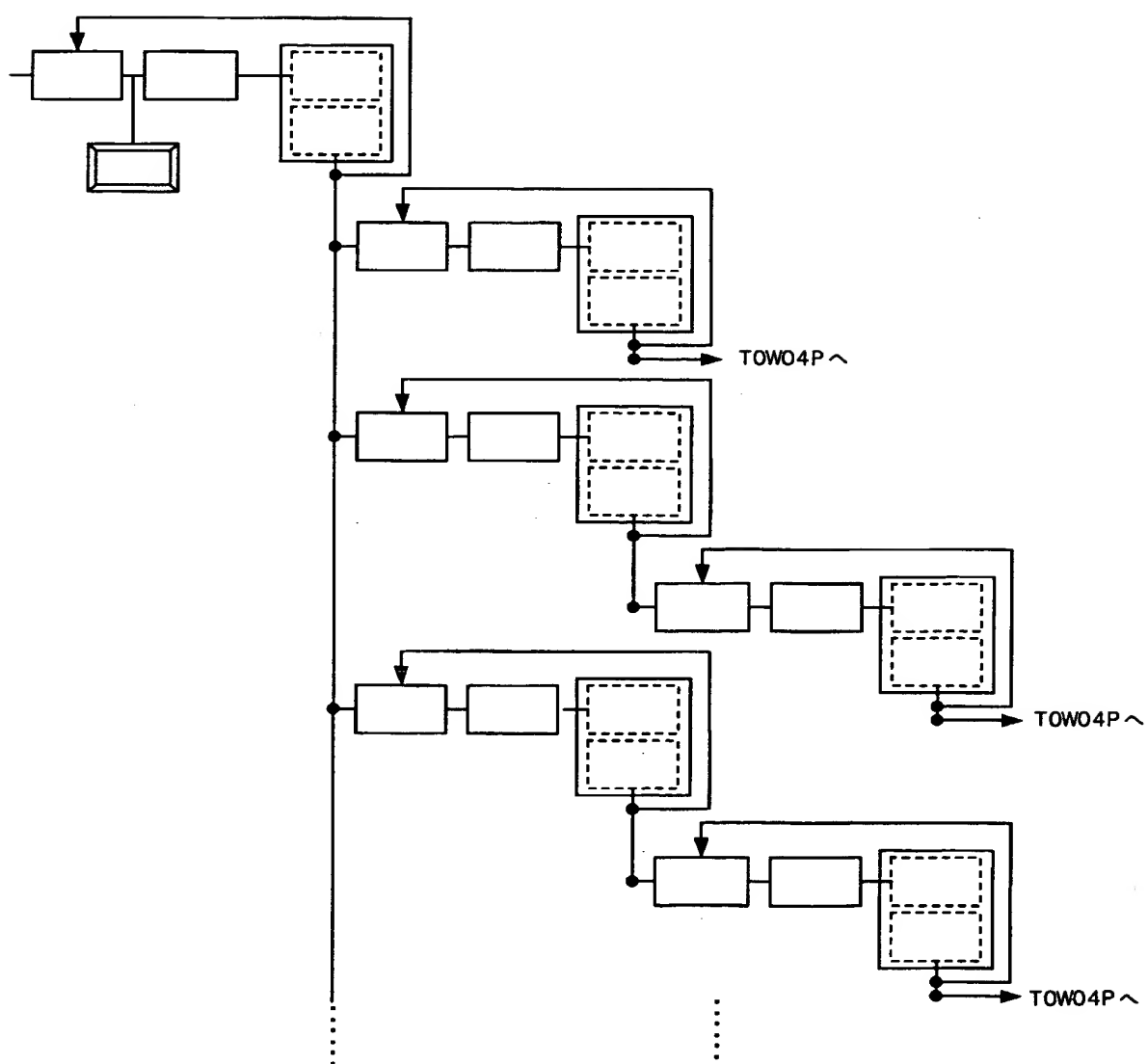
第4図



THIS PAGE BLANK (USPTO)

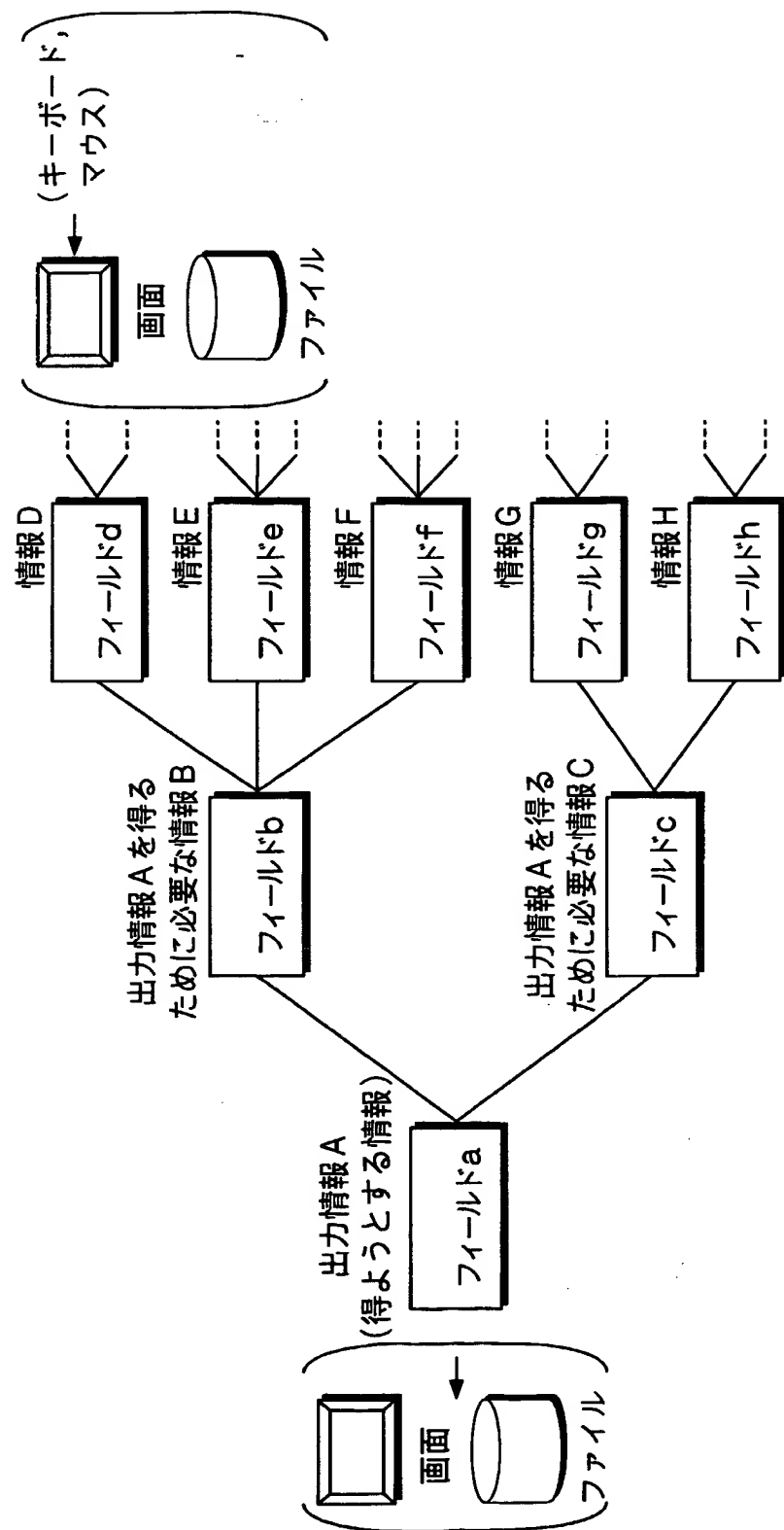
5 / 50

第5図



THIS PAGE BLANK (USPTO)

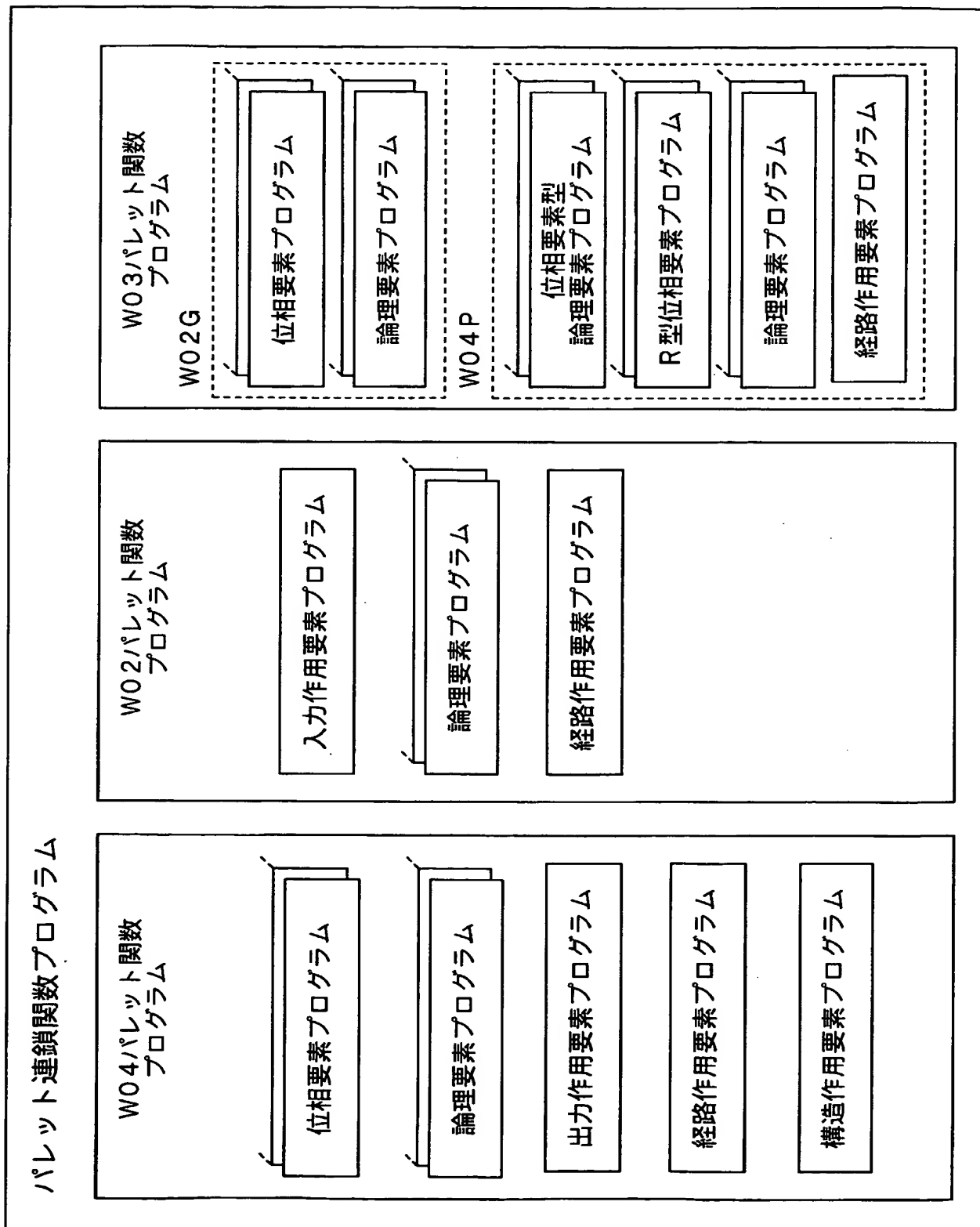
第6図



THIS PAGE BLANK (USPTO)

7 / 50

第7図

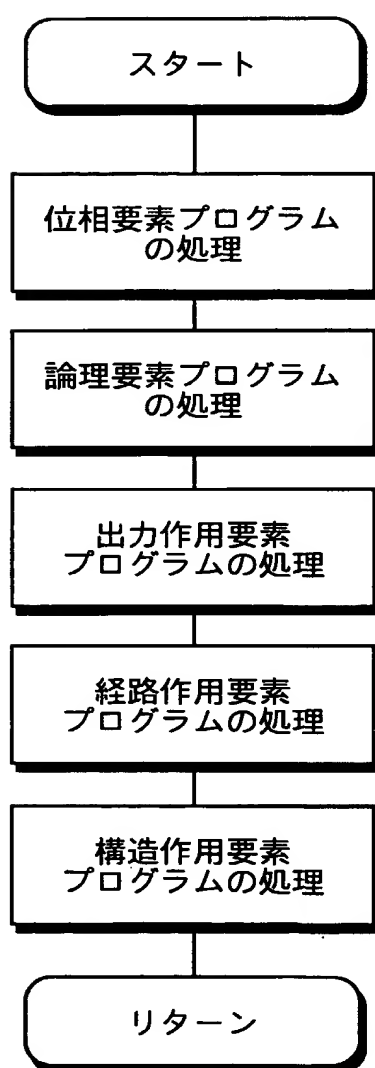


THIS PAGE BLANK (USPTO)

8 / 50

第8図

WO 4 パレット関数プログラム

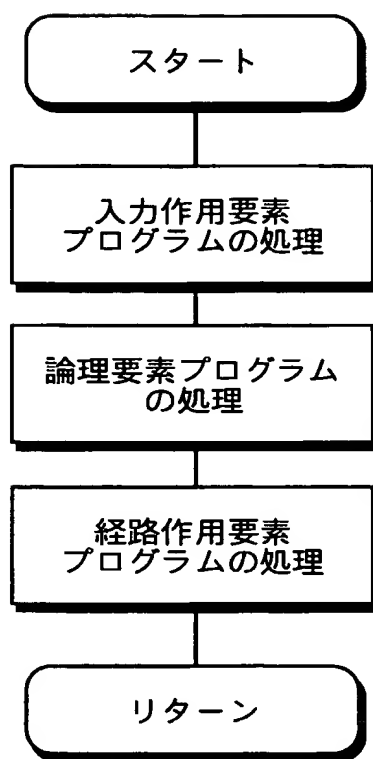


THIS PAGE BLANK (USPTO)

9 / 50

第9図

WO 2パレット関数プログラム

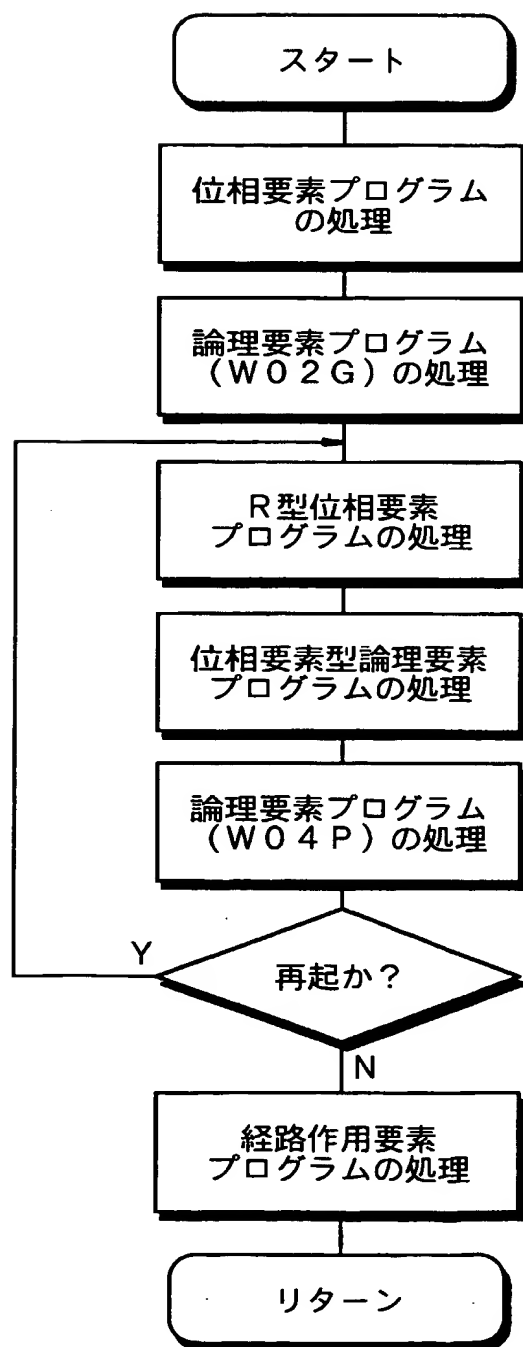


THIS PAGE BLANK (USPTO)

10/50

第10図

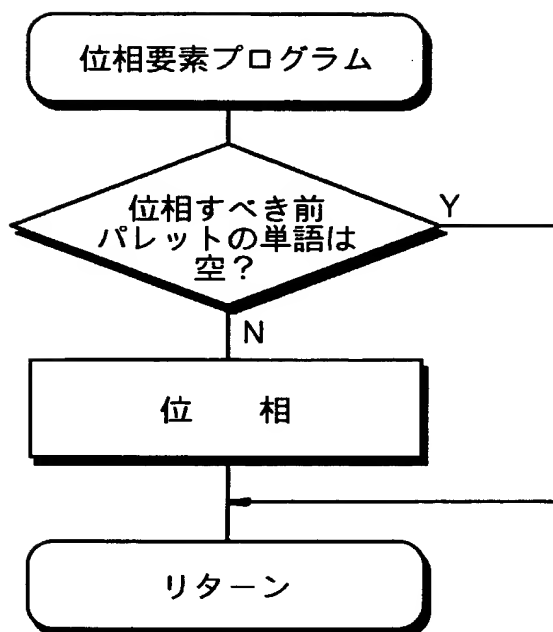
W03パレット関数プログラム



THIS PAGE BLANK (USPTO)

11 / 50

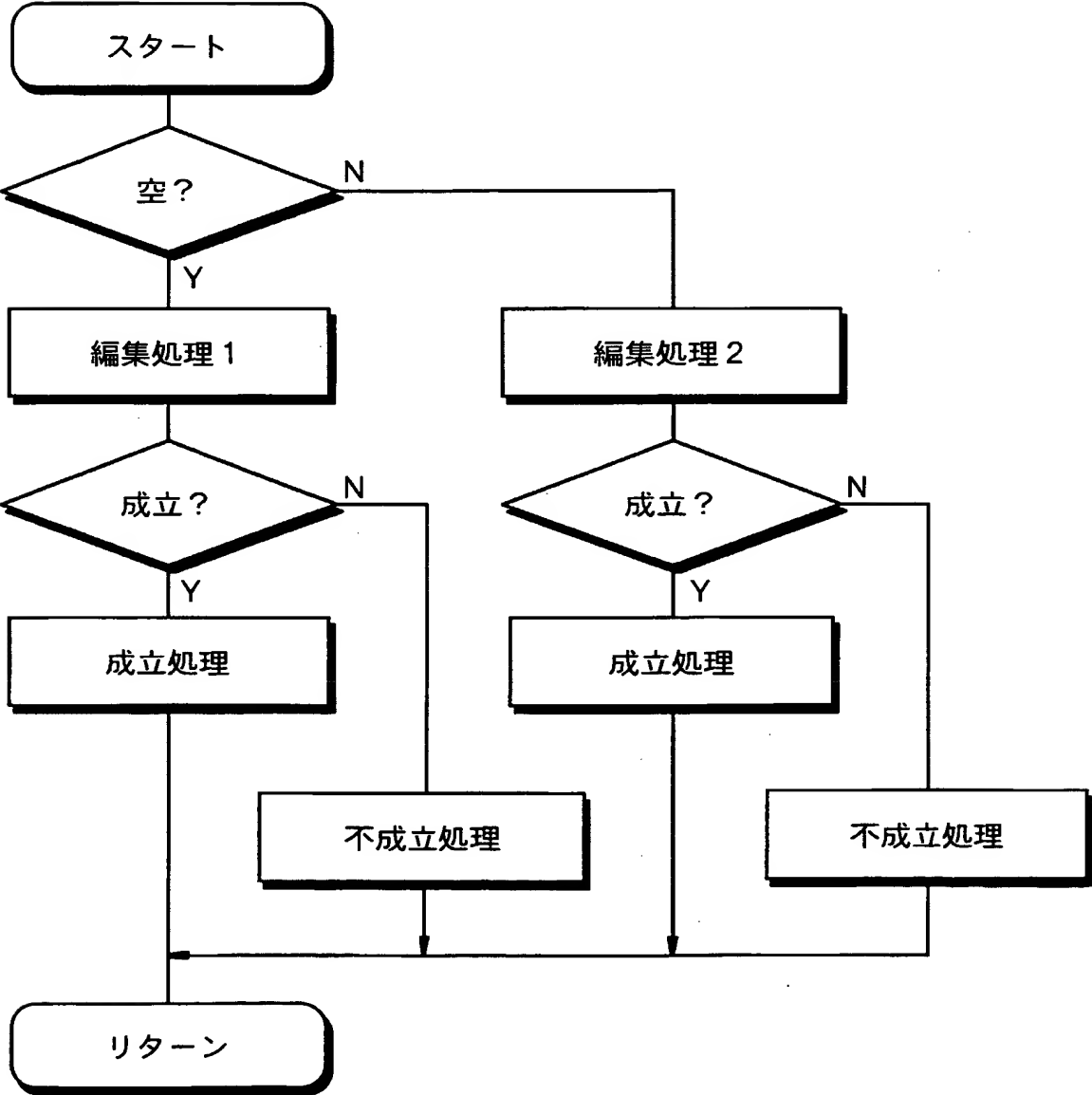
第 11 図



THIS PAGE BLANK (USPTO)

第 1 2 図

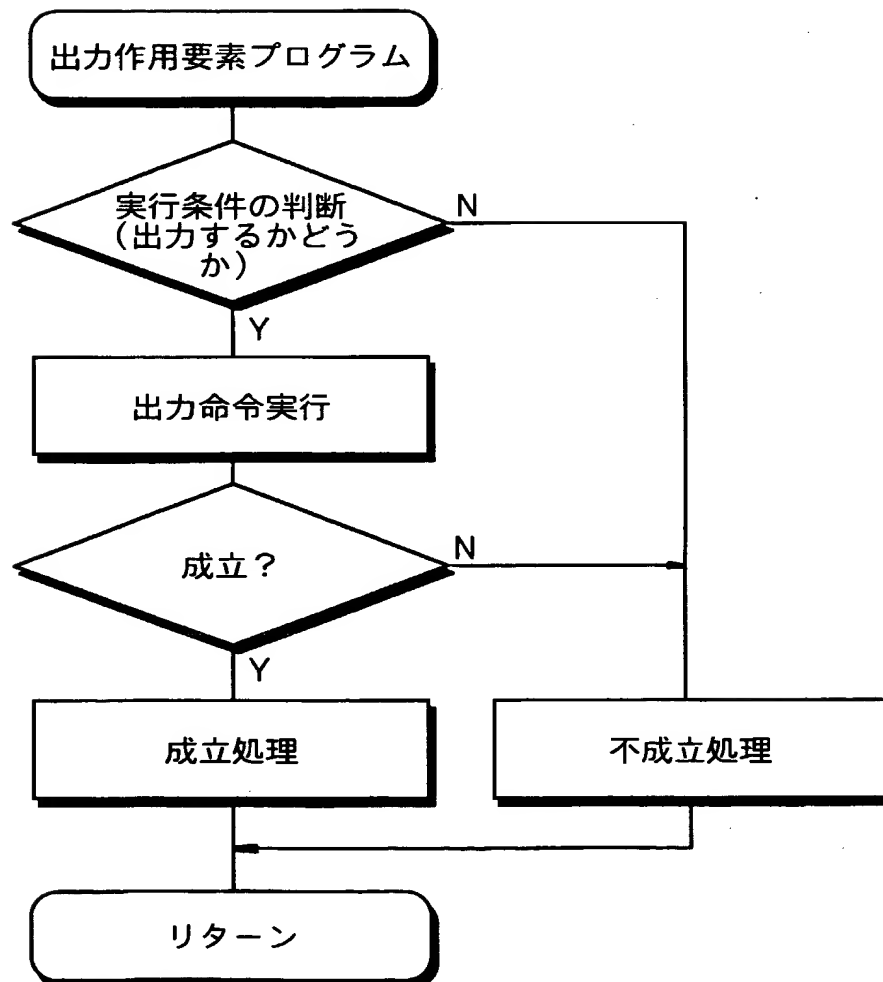
論理要素プログラム (W 0 4)



THIS PAGE BLANK (USPTO)

13/50

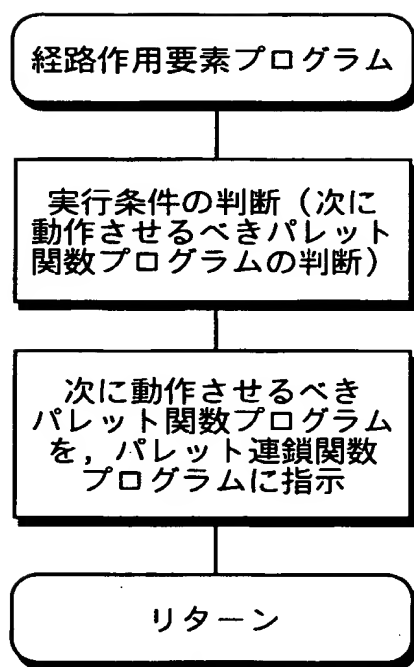
第13図



THIS PAGE BLANK (USPTO)

14 / 50

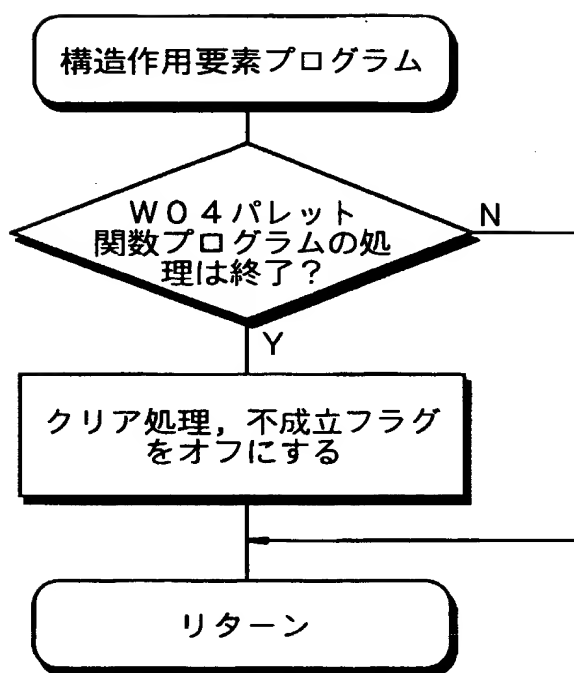
第14図



THIS PAGE BLANK (USPTO)

15 / 50

第15図

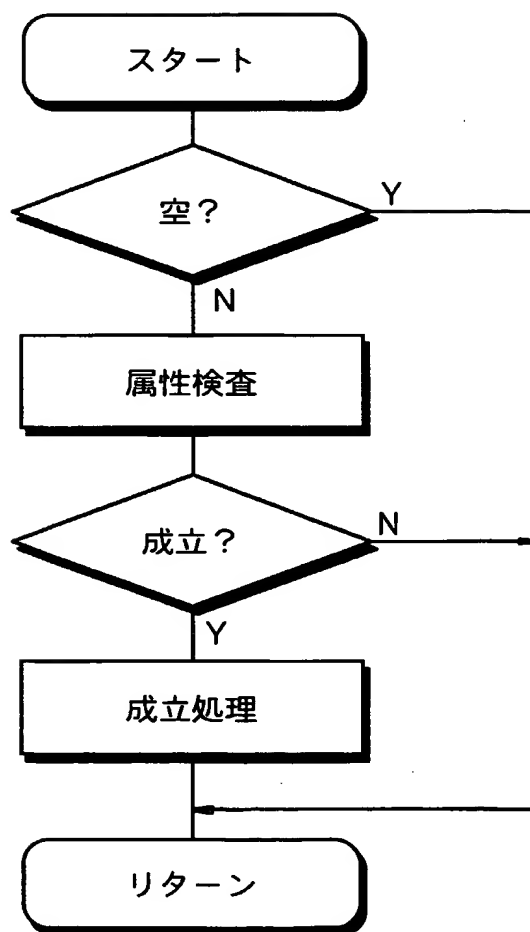


THIS PAGE BLANK (USPTO)

16 / 50

第16図

論理要素プログラム (W02)

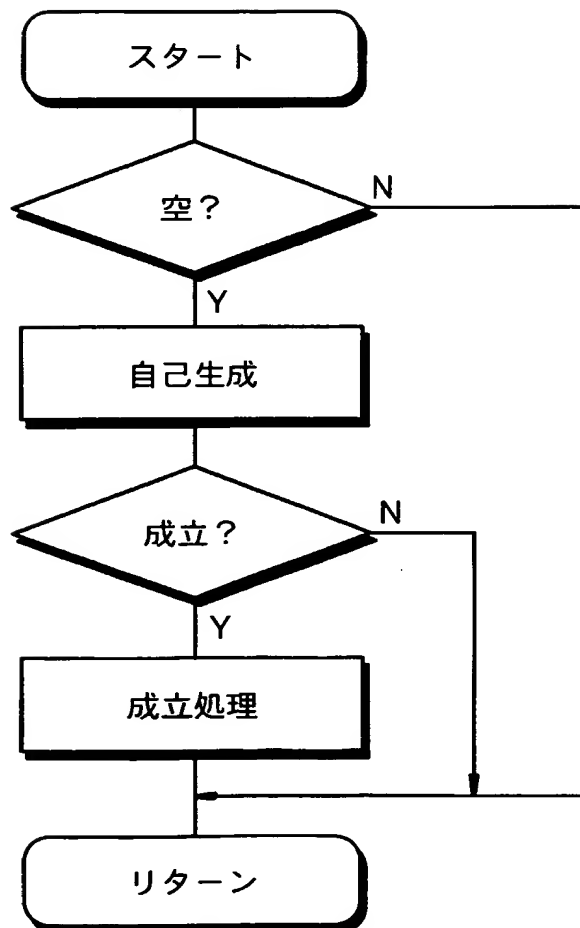


THIS PAGE BLANK (USPTO)

17/50

第17図

論理要素プログラム (W02G)



THIS PAGE BLANK (USPTO)

18 / 50

第 18 図

IF W02.□ = LOW-VALUE

GO TO EXIT.

END-IF

/*属性検査

IF W02.□ = NUMERIC

GO TO EXIT.

END-IF

IF W02.□ NOT = LOW-VALUE

GO TO EXIT.

END-IF

W02.□ CNT = W02.□_CNT+1

IF W02.□_CNT < W02_RECALL_MAX

W02_RECALL_FLG = "1"

ELSE

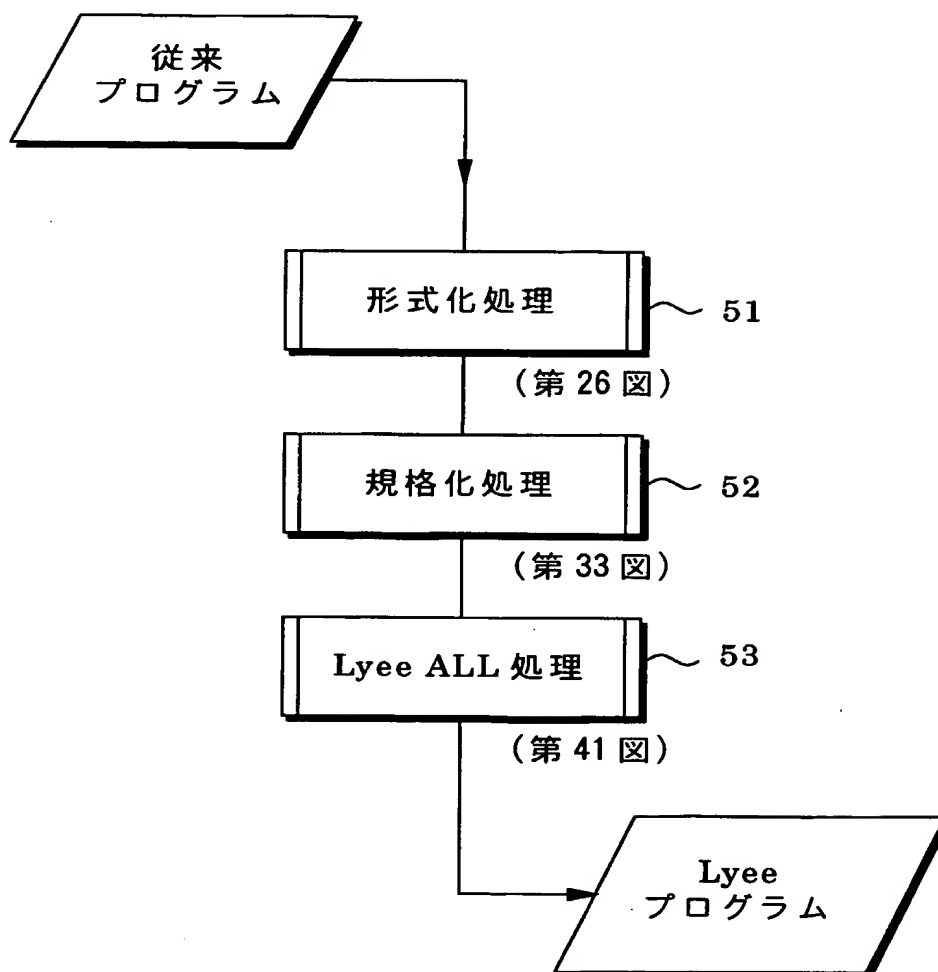
W02.□_Non = "1"

END-IF

THIS PAGE BLANK (USPTO)

19 / 50

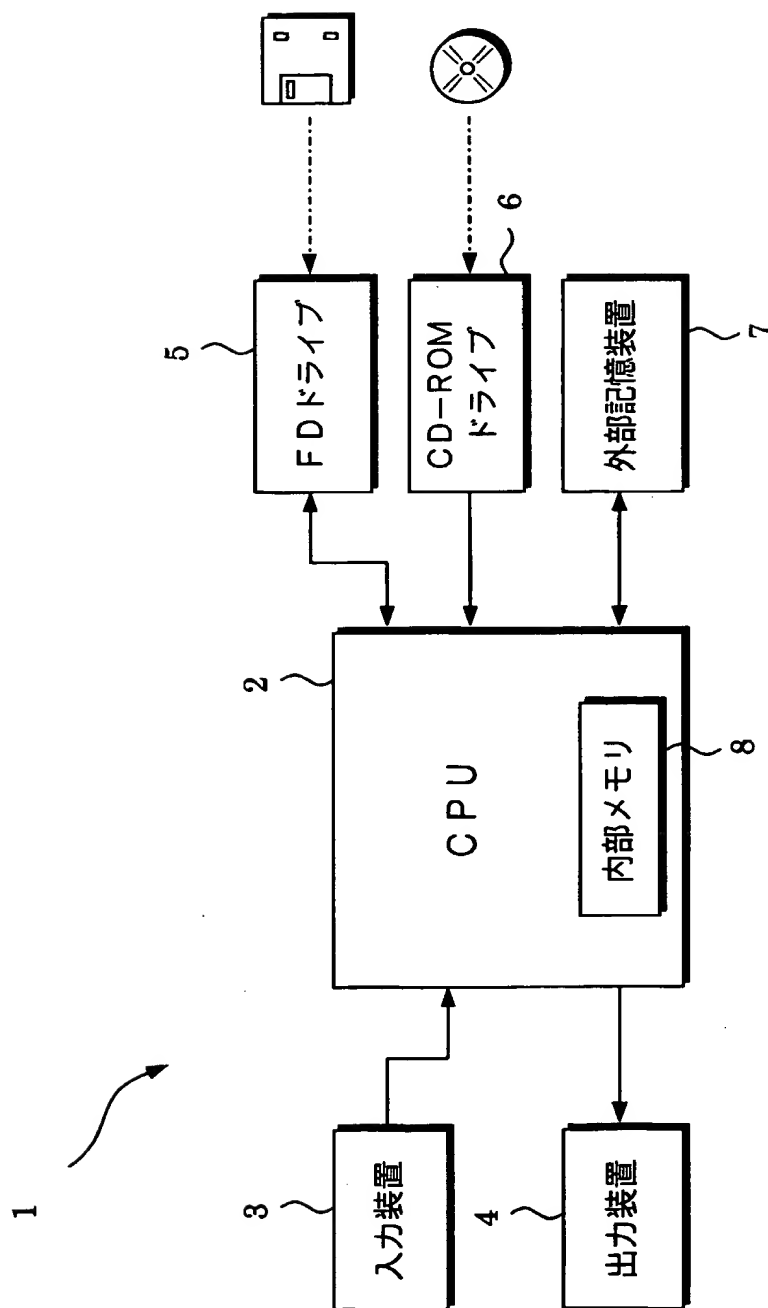
第 19 図



THIS PAGE BLANK (USPTO)

20/50

第20図



THIS PAGE BLANK (USPTO)

21 / 50

第 2 1 図

```
01  IDENTIFICATION DIVISION.
02  PROGRAM-ID.  AAA1.
03  ENVIRONMENT DIVISION.
04  CONFIGURATION SECTION.
05      SOURCE-COMPUTER.  AS400.
06      OBJECT-COMPUTER.  AS400.
07  INPUT-OUTPUT SECTION.
08  FILE-CONTROL.
09      SELECT GAMEN-F ASSIGN TO 画面-AAA.
10      ORGANIZATION IS TRANSACTION.

11  DATA DIVISION.
12  FILE SECTION.
13  FD  GAMEN-F.
14      01  GAMEN-R.
15          03  SHIN-CD      PIC  X(05).
16          03  SU           PIC  S9(02).
17          03  TANKA        PIC  S9(05).
18          03  KINGAKU      PIC  S9(05).

19  WORKING-STRAGE SECTION.
20      01  SHIN-TBL.
21          03  SHIN-CD      PIC  X(05).
22          03  TANKA        PIC  S9(05).
23      01  END-BTN          PIC  X(01).
24      01  WK               PIC  S9(05).

01  PROCEDURE DIVISION.
02  MAIN-AA SECTION.
03  MAIN-START.
04      OPEN I-O      GAMEN-F.
05      INITIALIZE GAMEN-R.
06      WRITE GAMEN-R.
07  LOOP-1.
08      READ GAMEN-F.
09      IF END-BTN = "1"
10          CLOSE GAMEN-F
11          GO TO MAIN-EXIT
12      END-IF.
13      IF SHIN-CD OF GAMEN-R = SPACE OR SU OF GAMEN-R = ZERO
14          MOVE 99999 TO KINGAKU OF GAMEN-R
15      ELSE
16          MOVE SHIN-CD OF GAMEN-R TO SHIN-CD OF SHIN-TBL.
17          SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18          IF ステータス NOT = ZERO
19              MOVE 99999 TO TANKA OF GAMEN-R
20          ELSE
21              MOVE TANKA OF SHIN-TBL TO TANKA OF GAMEN-R
22              COMPUTE WK = TANKA OF GAMEN-R * SU OF GAMEN-R
23              IF WK > 10000
24                  COMPUTE KINGAKU OF GAMEN-R = WK * 0.8
25              ELSE
26                  COMPUTE KINGAKU OF GAMEN-R = WK * 0.9
27          END-IF
28      END-IF.
29  END-IF.
30  WRITE GAMEN-R.
31  GO TO LOOP-1.
32  MAIN-EXIT.
33  STOP RUN.
```

THIS PAGE BLANK (USPTO)

22 / 50

第22図

54

商品受注画面

〈商品受注画面〉

商品コード	AAAAA	55
数 量	20	56
単 価	100	57
金 額	2000	58
	59	終了

THIS PAGE BLANK (USPTO)

23 / 50

第23図

SHIN-TBL (商品テーブル)

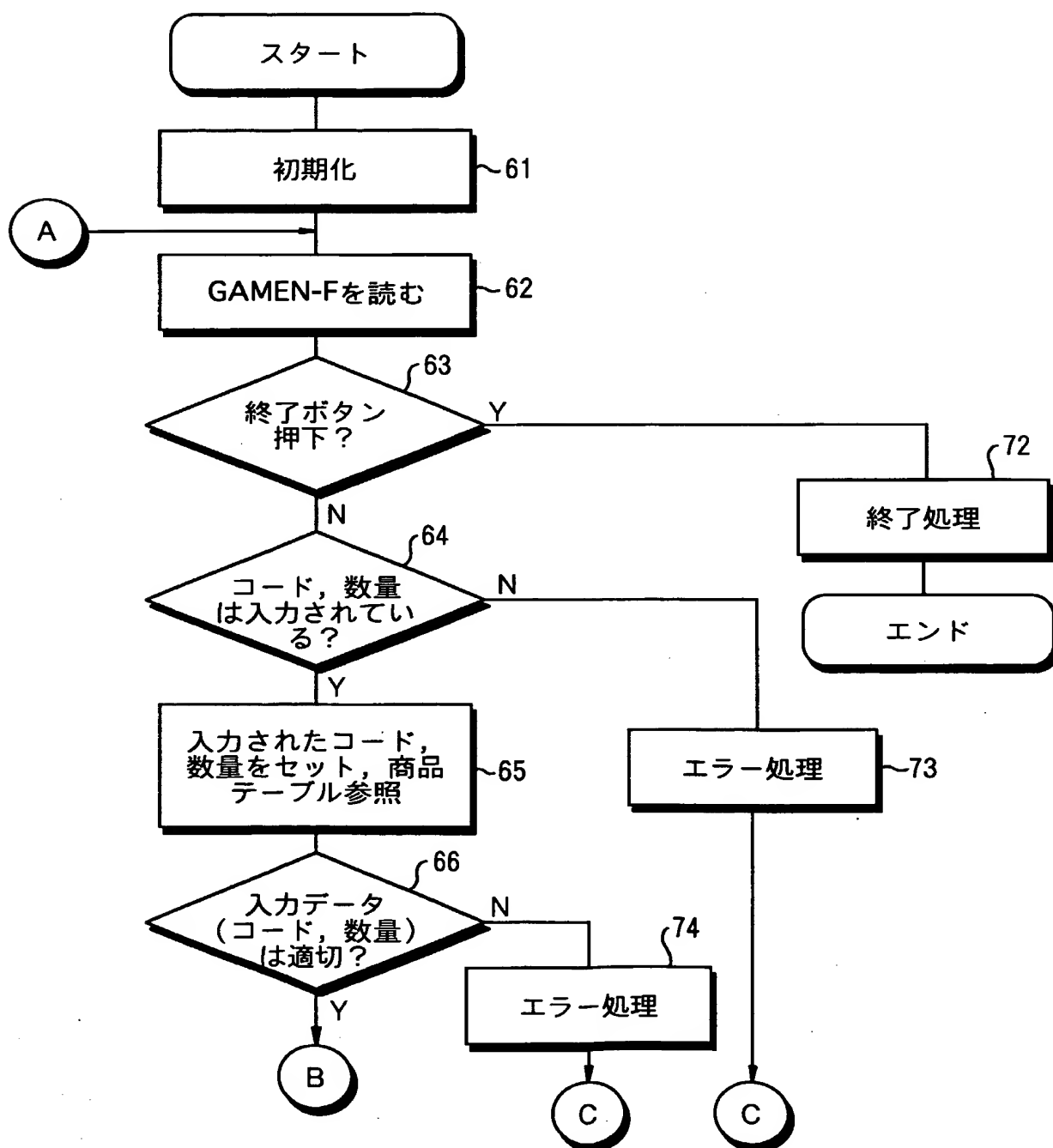
SHIN-CD (商品コード)	TANKA (単価)
AAAAA	100
BBBBB	200
CCCCC	300
DDDDD	400
EEEEE	500
⋮	⋮

60

THIS PAGE BLANK (USPTO)

24 / 50

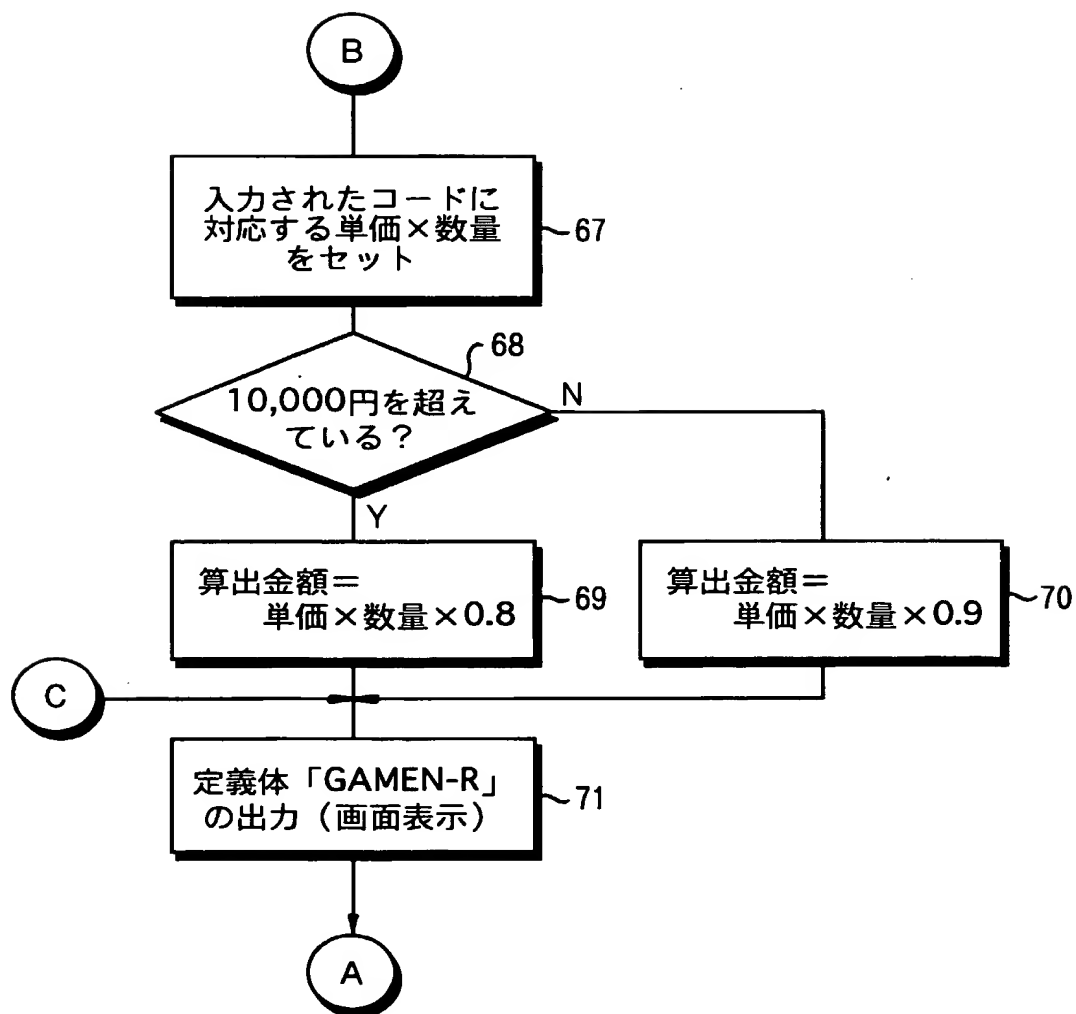
第24図



THIS PAGE BLANK (USPTO)

25 / 50

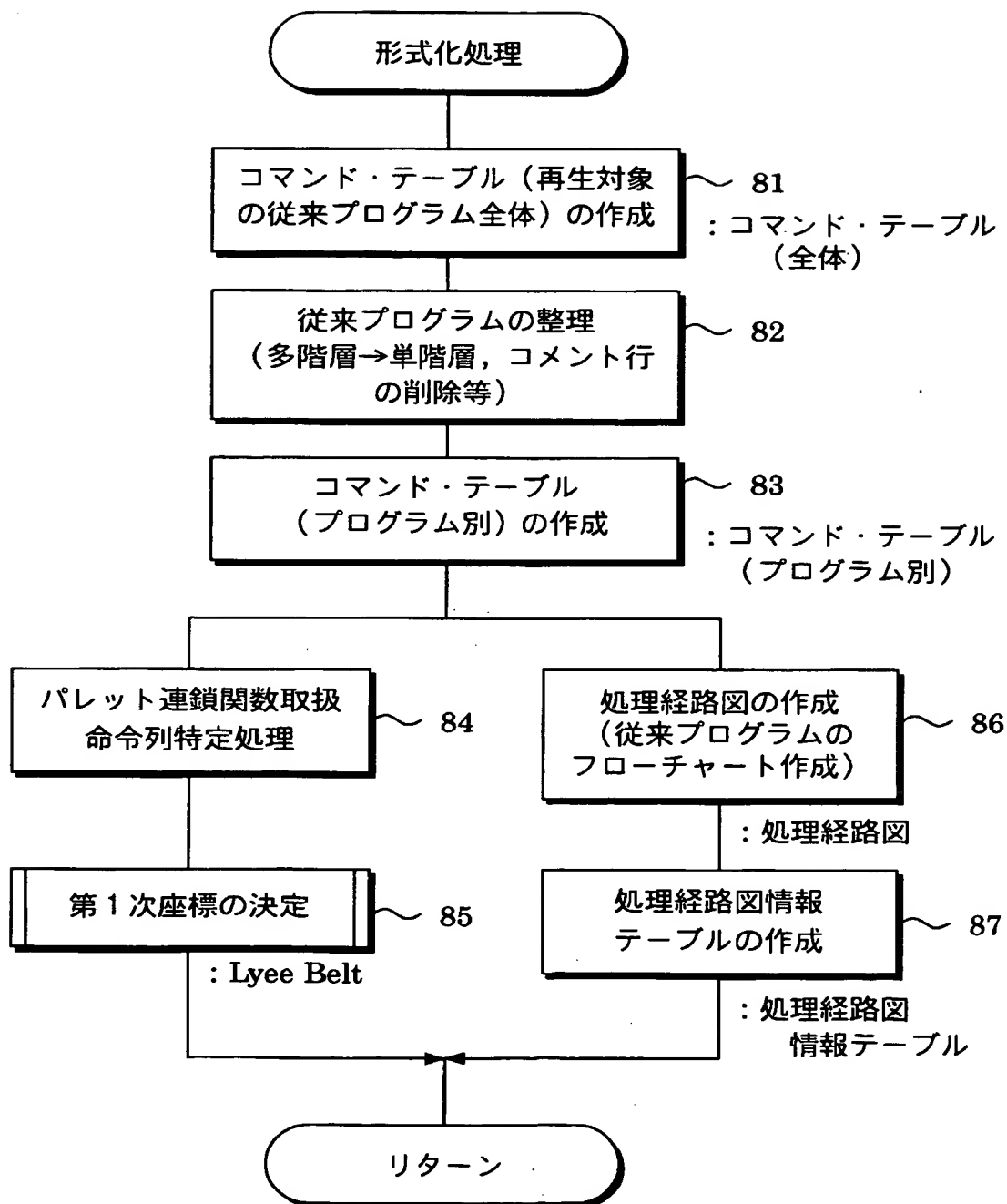
第 25 図



THIS PAGE BLANK (USPTO)

26 / 50

第26図



THIS PAGE BLANK (USPTO)

第 27 図

コマンド・テーブル(全体)

定義体 種別	コマンド	命令 種別	条件	入出力装置	入/出	ハレット 所在
画面	SELECT	COBOL	ASSIGN TO 画面	ASSIGN文節 ファイル名	入	TOW02
画面	SELECT	COBOL	ASSIGN TO 画面	ASSIGN文節 ファイル名	出	TOW04
DB	SELECT	COBOL	FROM	FROM文節 DB名	入	T1W02

THIS PAGE BLANK (USPTO)

第28図

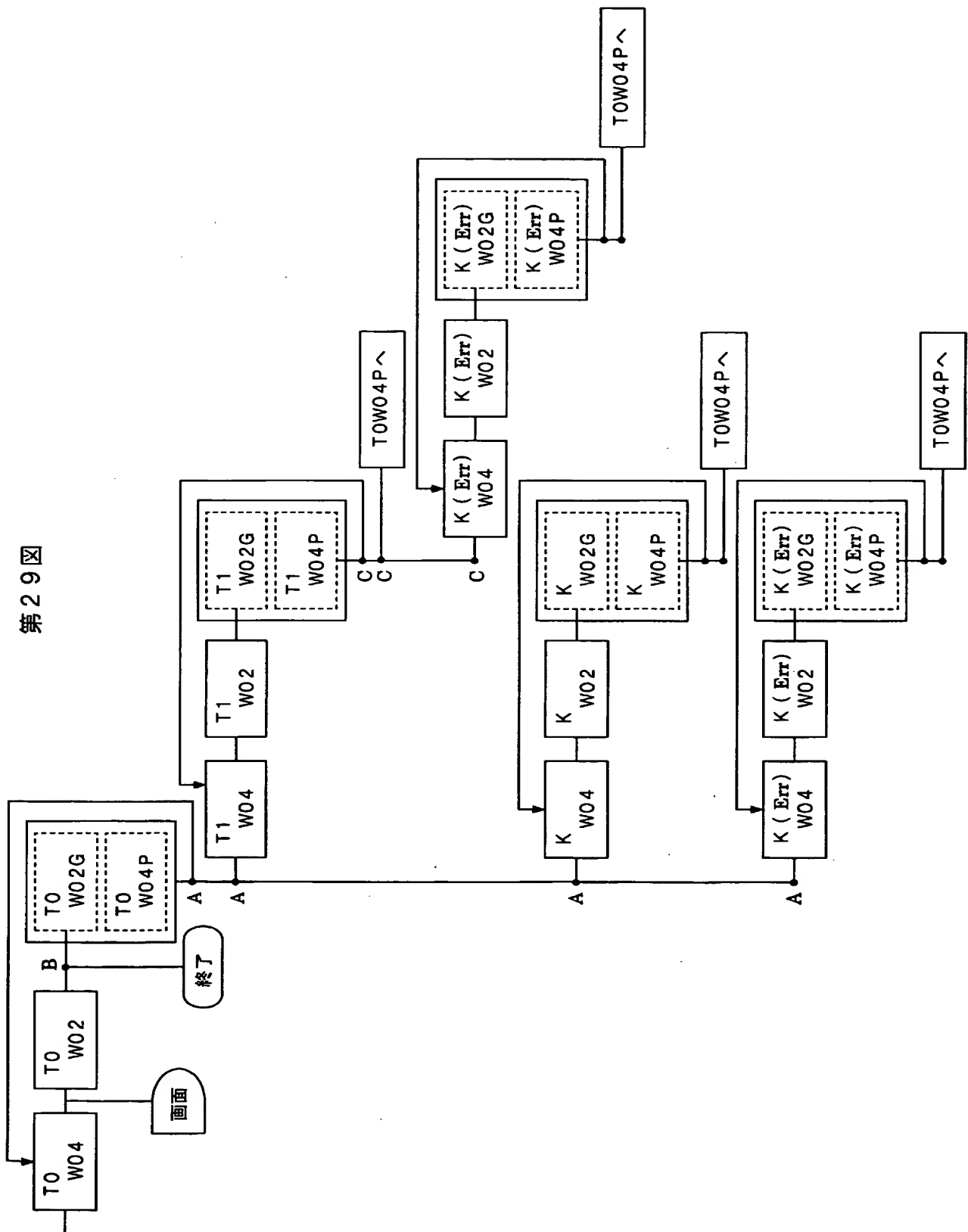
コマンドテーブル(プログラム別)

プログラム ID	コマンド	入出力装置	定義体	入/出	ハレット 所在
AAA1	READ	AAA	GAMEN-R	入	TOW02
AAA1	WRITE	AAA	GAMEN-R	出	TOW04
AAA1	SELECT	SHIN-DB	SHIN-TBL	入	T1W02

THIS PAGE BLANK (USPTO)

29/50

第29図



THIS PAGE BLANK (USPTO)

30/50

第30図

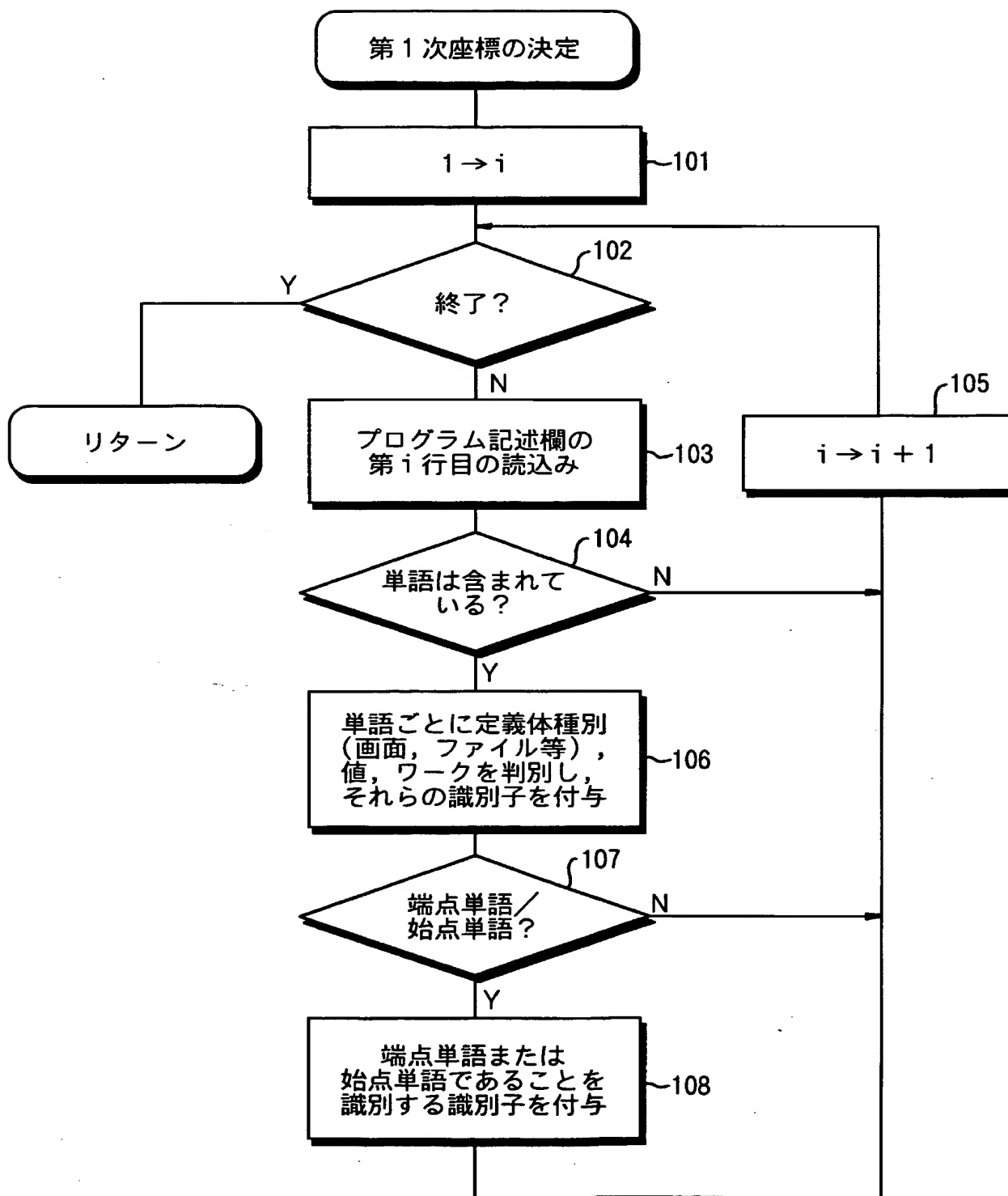
処理経路図情報テーブル

現PGM ID	処理経路図 ID	連鎖関数 ID	ハレット ID	経路要素 ID	次ハレット ID	入出力コメント 要素ID	コメント	定義体
A1	Sample1	Sample_1	A1TOW04	Rute-1	A1TOW02	WRITE-1	WRITE	GAMEN-R
A1	Sample1	Sample_1	A1TOW02	Rute-1	A1TOW03			
				Rute-2	STOP			
A1	Sample1	Sample_1	A1TOW03	Rute-1	A1TOW04	READ-1	READ	GAMEN-R
				Rute-2	A1T1W04			
				Rute-3	A1K1W04			
				Rute-4	A1E1W04			
A1	Sample2	Sample_2	A1T1W04	Rute-1	A1T1W02			
A1	Sample2	Sample_2	A1T1W02	Rute-1	A1T1W03			
A1	Sample2	Sample_2	A1T1W03	Rute-1	A1TOW04P	READ-1	SELECT	SHIN-DB
				Rute-2	A1E2W04			
A1	Sample3	Sample_3	A1K1W04	Rute-1	A1K1W02			
A1	Sample3	Sample_3	A1K1W02	Rute-1	A1K1W03			
A1	Sample3	Sample_3	A1K1W03	Rute-1	A1TOW04P			
A1	Sample4	Sample_4	A1E1W04	Rute-1	A1E1W02			
A1	Sample4	Sample_4	A1E1W02	Rute-1	A1E1W03			
A1	Sample4	Sample_4	A1E1W03	Rute-1	A1TOW04P			
A1	Sample5	Sample_5	A1E2W04	Rute-1	A1E2W02			
A1	Sample5	Sample_5	A1E2W02	Rute-1	A1E2W03			
A1	Sample5	Sample_5	A1E2W03	Rute-1	A1TOW04P			

THIS PAGE BLANK (USPTO)

31 / 50

第31図



THIS PAGE BLANK (USPTO)

32/50

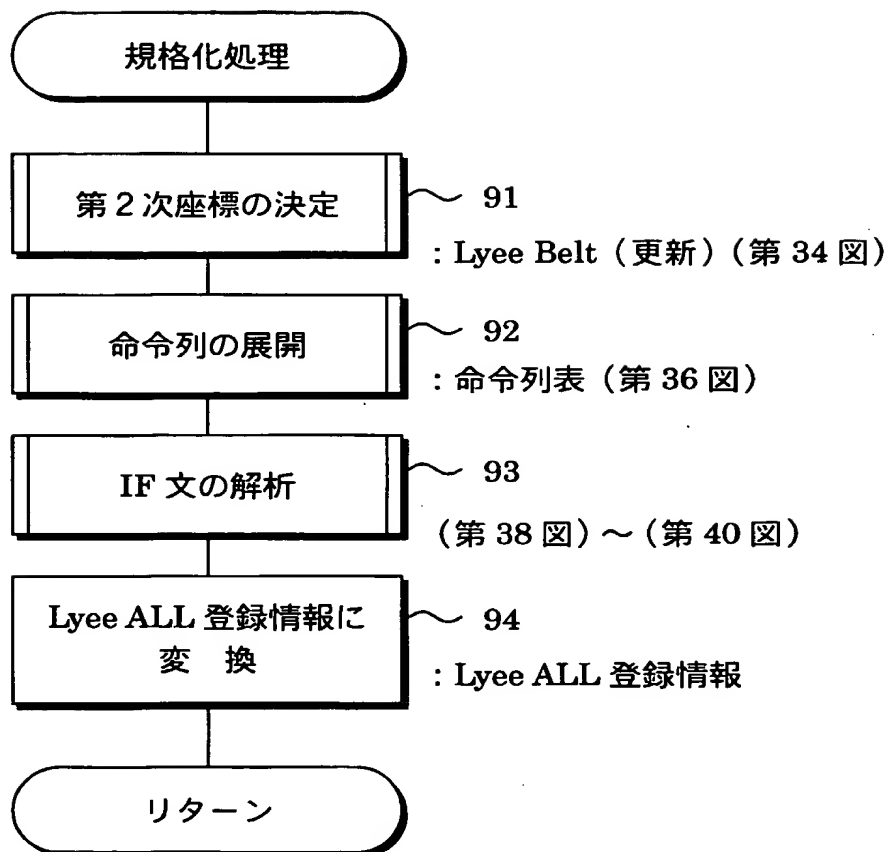
第32図

行 番 号	命 令 種 別	連 鎖 関 数	プログラム記述
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1.
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN (画面) = "1" (値)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R (画面) = SPACE (値)
			OR SU OF GAMEN-R (画面) = ZERO (値)
14	端点命令		MOVE 99999 (始: 値) TO KINGAKU OF GAMEN-R (端: 画面)
15			ELSE
16	端点命令		MOVE SHIN-CD OF GAMEN-R (始: 画面)
			TO SHIN-CD OF SHIN-TBL. (端: 制御BOX)
17	CMD		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18	IF		IF ステータス (制御BOX) NOT = ZERO (値)
19	端点命令		MOVE 99999 (始: 値) TO TANKA OF GAMEN-R (端: 画面)
20			ELSE
21	端点命令		MOVE TANKA OF SHIN-TBL (始: DB)
			TO TANKA OF GAMEN-R (端: 画面)
22	端点命令		COMPUTE WK (端: ワ-ク) = TANKA OF GAMEN-R (始: 画面)
			* SU OF GAMEN-R (始: 画面)
23	IF		IF WK (ワ-ク) > 10000 (値)
24	端点命令		COMPUTE KINGAKU OF GAMEN-R (端: 画面) = WK (始: ワ-ク) * 0.8
25			ELSE
26			COMPUTE KINGAKU OF GAMEN-R (端: 画面) = WK (始: ワ-ク) * 0.9
27			END-IF
28			END-IF
29			END-IF.
30		Φ	WRITE GAMEN-R.
31			GO TO LOOP-1.
32			MAIN-EXIT.
33			STOP RUN.

THIS PAGE BLANK (USPTO)

33 / 50

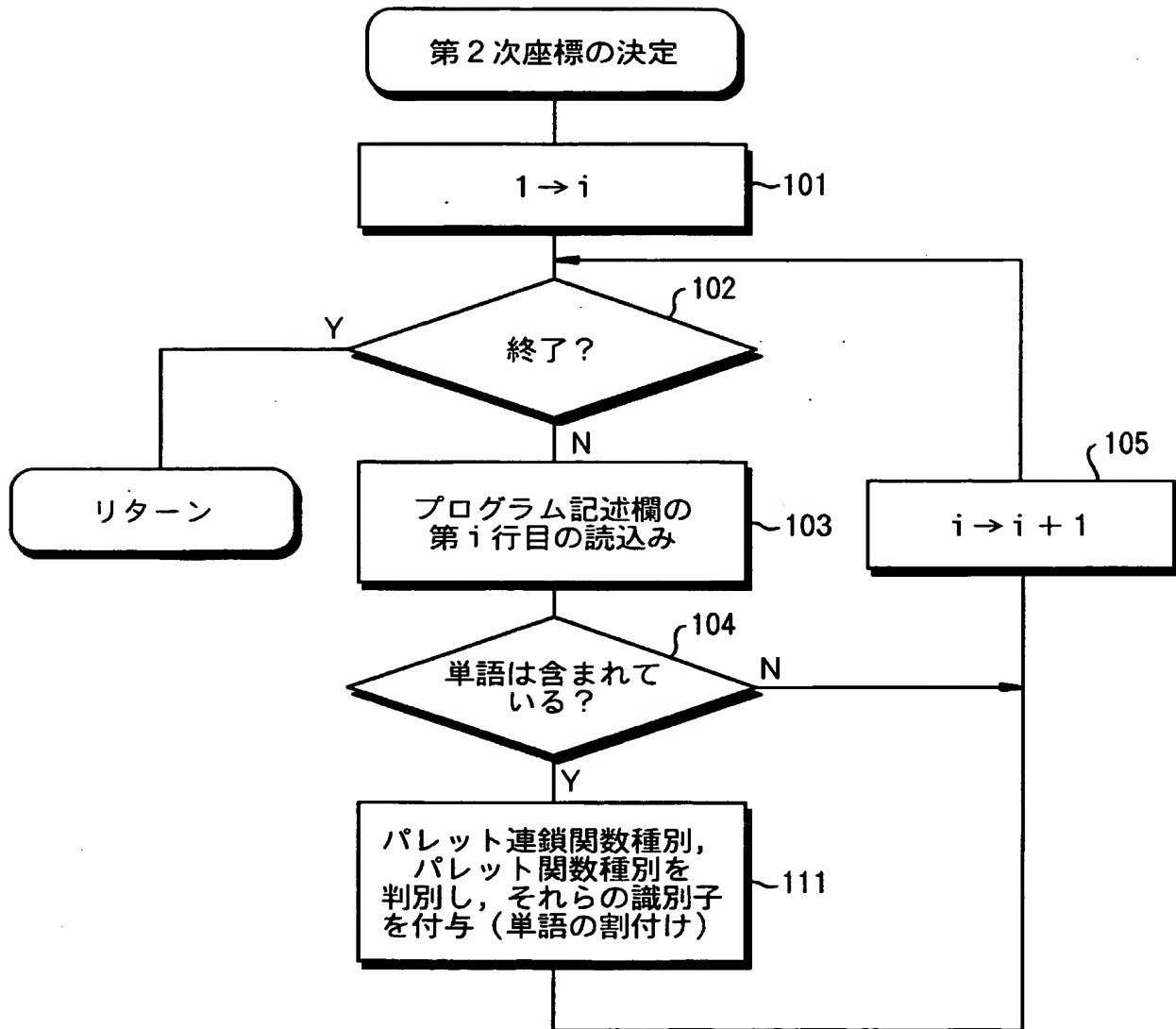
第 33 図



THIS PAGE BLANK (USPTO)

34 / 50

第34図



THIS PAGE BLANK (USPTO)

35 / 50

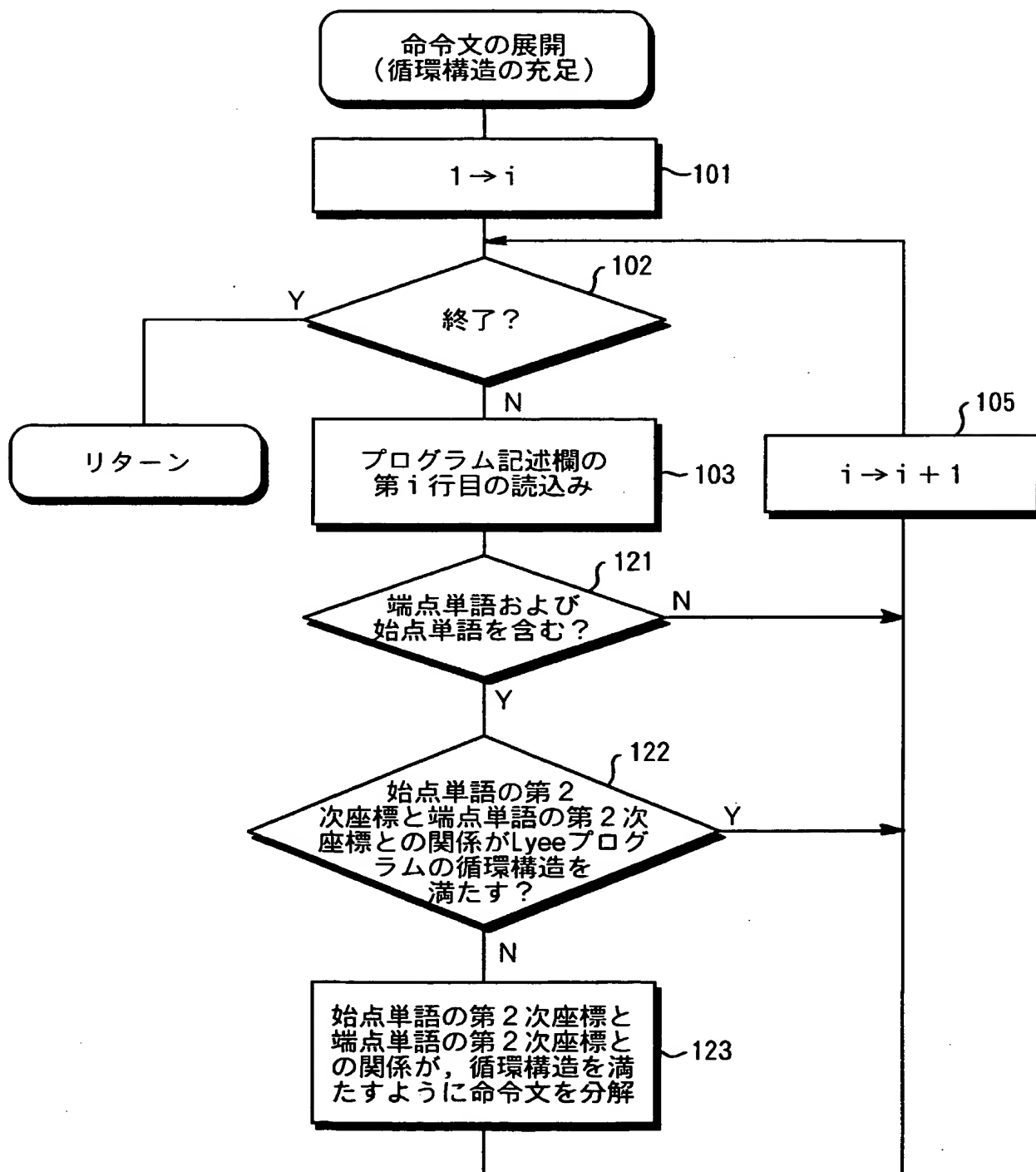
第35図

行 番 号	命 令 種 別	連 鎖 関 数	プログラム記述
01			PROCEDURE DIVISION.
02			MAIN-AA SECTION.
03			MAIN-START.
04		Φ	OPEN I-O GAMEN-F.
05		Φ	INITIALIZE GAMEN-R.
06		Φ	WRITE GAMEN-R.
07			LOOP-1.
08		Φ	READ GAMEN-F.
09	IF		IF END-BTN (TOW02) = "1" (値)
10		Φ	CLOSE GAMEN-F
11			GO TO MAIN-EXIT
12			END-IF.
13	IF		IF SHIN-CD OF GAMEN-R (TOW02) = SPACE (値)
			OR SU OF GAMEN-R (TOW02) = ZERO (値)
14	端点命令		MOVE 99999 (始: 値) TO KINGAKU OF GAMEN-R (端: TOW04)
15			ELSE
16	端点命令		MOVE SHIN-CD OF GAMEN-R (始: TOW02)
			TO SHIN-CD OF SHIN-TBL. (端: T1CB)
17	CMD		SELECT TEIKA FROM SHIN-DB INTO :SHIN-TBL.TANKA
18	IF		IF ステータス (T1CB) NOT = ZERO (値)
19	端点命令		MOVE 99999 (始: 値) TO TANKA OF GAMEN-R (端: TOW04)
20			ELSE
21	端点命令		MOVE TANKA OF SHIN-TBL (始: T1W02)
			TO TANKA OF GAMEN-R (端: TOW04)
22	端点命令		COMPUTE WK (端: KW04P) = TANKA OF GAMEN-R (始: TOW04)
			* SU OF GAMEN-R (始: TOW02)
23	IF		IF WK (KW04P) > 10000 (値)
24	端点命令		COMPUTE KINGAKU OF GAMEN-R (端: TOW04) = WK (始: KW04P) * 0.8
25			ELSE
26			COMPUTE KINGAKU OF GAMEN-R (端: TOW04) = WK (始: KW04P) * 0.9
27			END-IF
28			END-IF
29			END-IF.
30		Φ	WRITE GAMEN-R.
31			GO TO LOOP-1.
32			MAIN-EXIT.
33			STOP RUN.

THIS PAGE BLANK (USPTO)

36 / 50

第36図



THIS PAGE BLANK (USPTO)

37 / 50

第37図

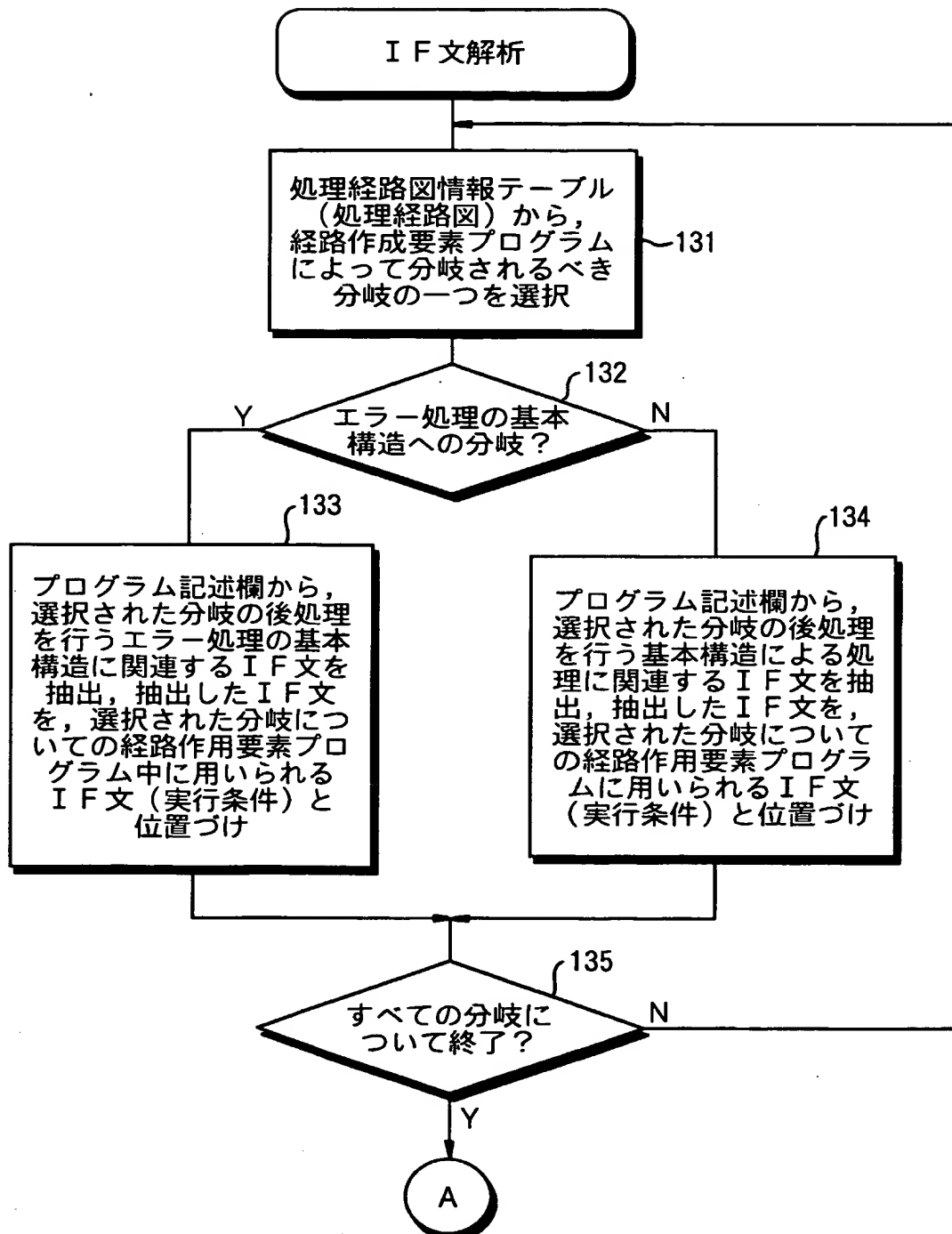
命令列表

行 番 号	種 別	所 在	展開後命令文
21	位相要素	T1W02G	MOVE TANKA OF SHIN-TBL(T1W02) TO TANKA OF SHIN-TBL(T1W02G)
21	位相要素型 論理要素	T1W04P	MOVE TANKA OF SHIN-TBL(T1W02G) TO TANKA OF SHIN-TBL(T1W04P境界)
21	R型位相要素	TOW04P	MOVE TANKA OF SHIN-TBL(T1W04P境界) TO TANKA OF SHIN-TBL(TOW04P境界)
21	論理要素	TOW04P	MOVE TANKA OF SHIN-TBL(TOW04P境界) TO TANKA OF GAMEN-R(TOW04P)
21	位相要素	TOW04	MOVE TANKA OF GAMEN-R(TOW04P) TO TANKA OF GAMEN-R(TOW04)

THIS PAGE BLANK (USPTO)

38/50

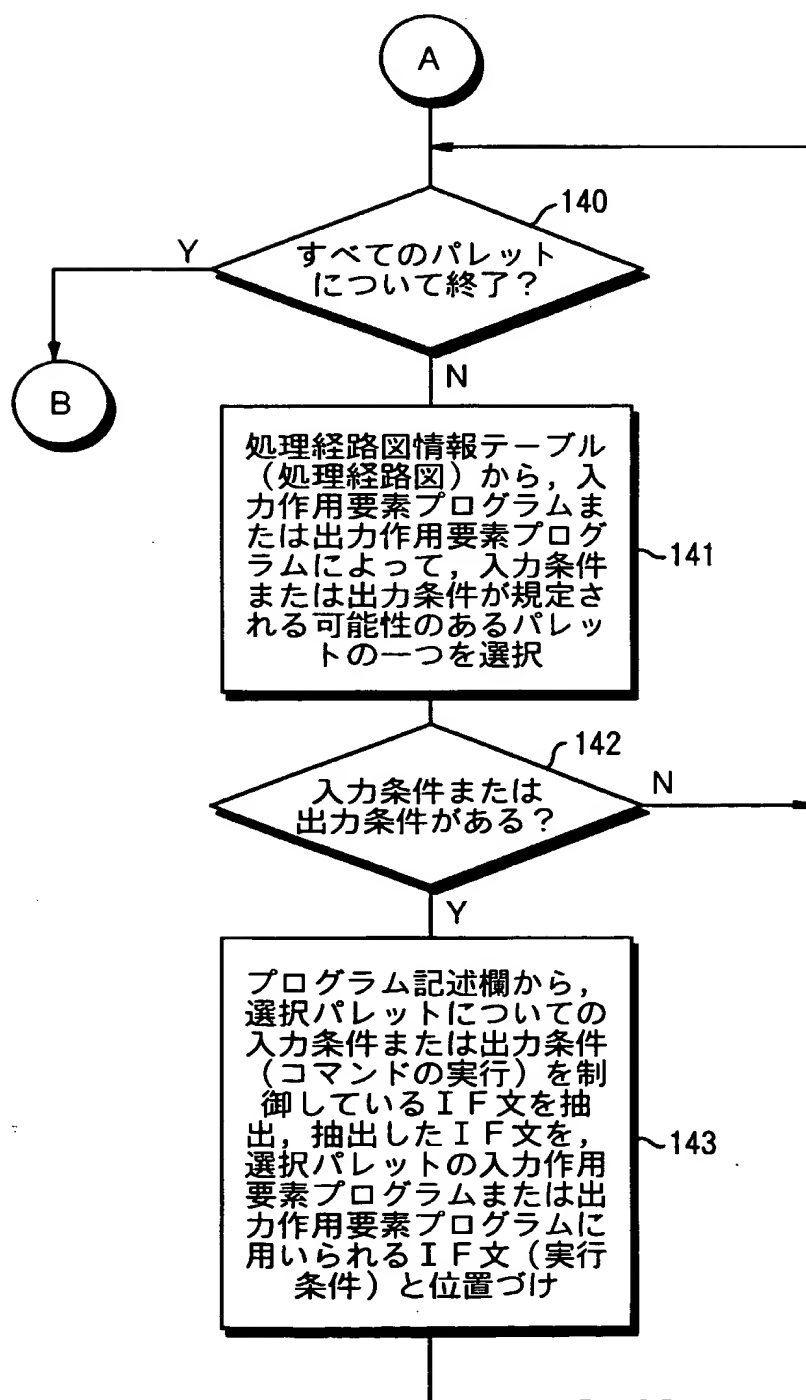
第38図



THIS PAGE BLANK (USPTO)

39/50

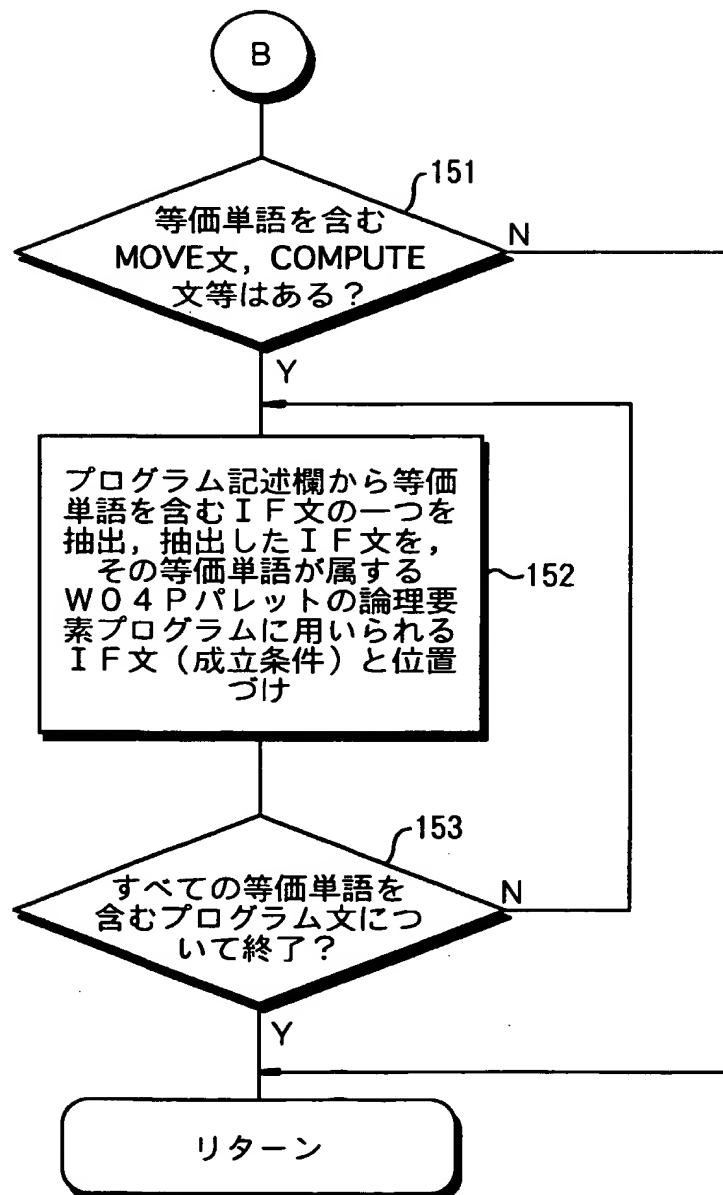
第39図



THIS PAGE BLANK (USPTO)

40/50

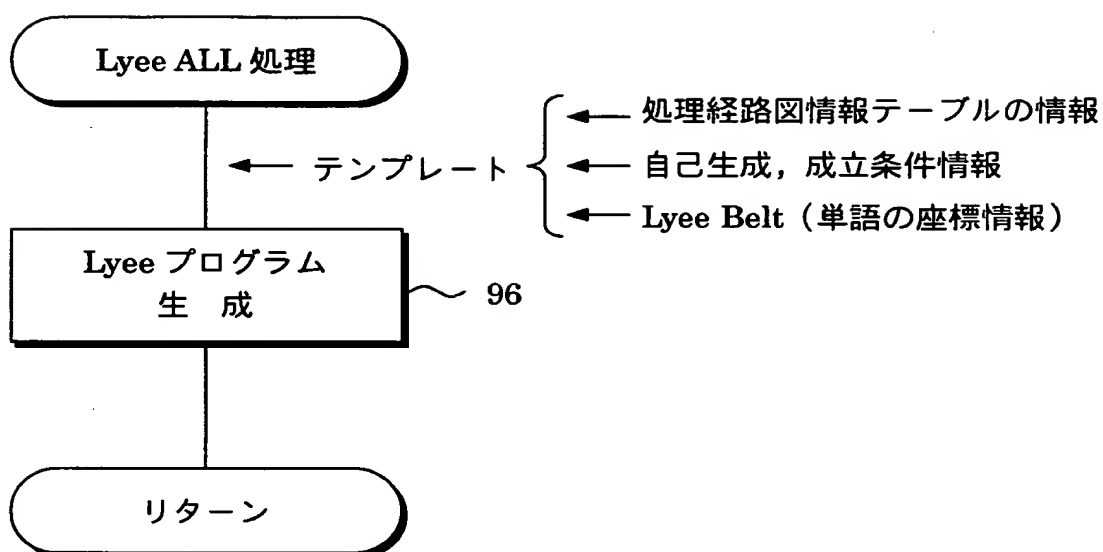
第40図



THIS PAGE BLANK (USPTO)

41 / 50

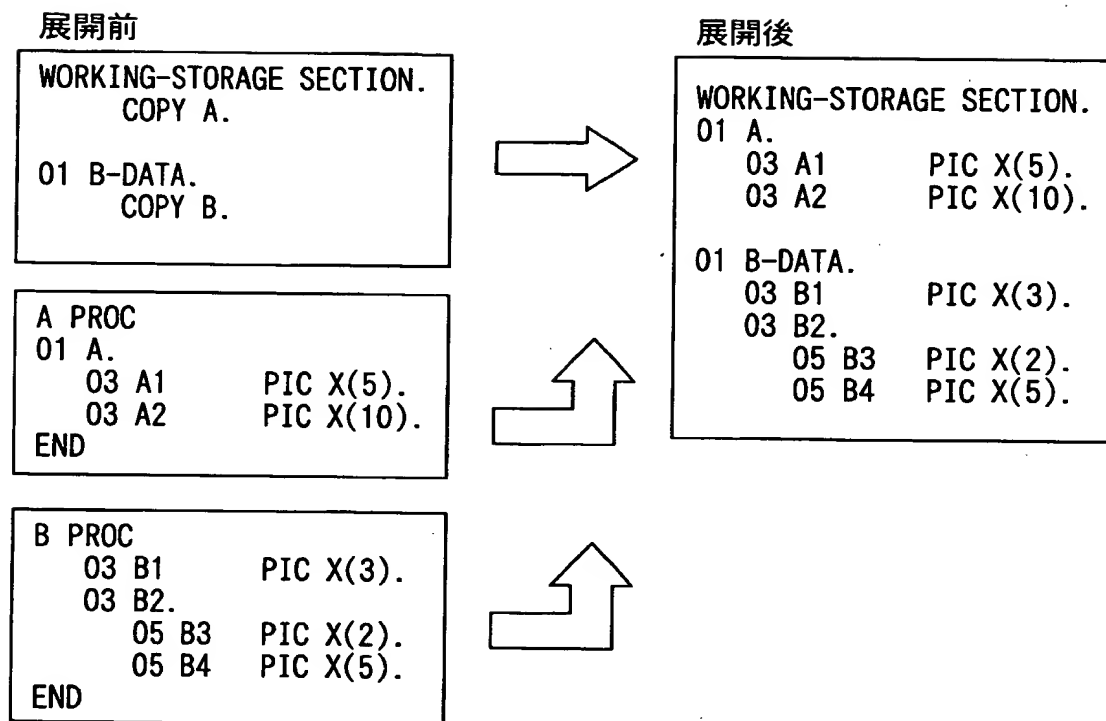
第41図



THIS PAGE BLANK (USPTO)

42 / 50

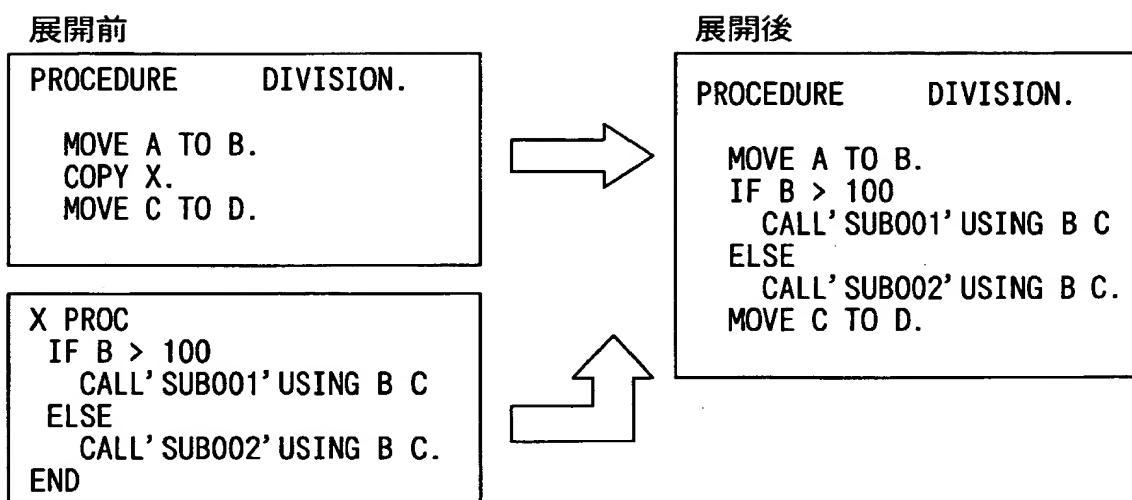
第42図



THIS PAGE BLANK (USPTO)

43 / 50

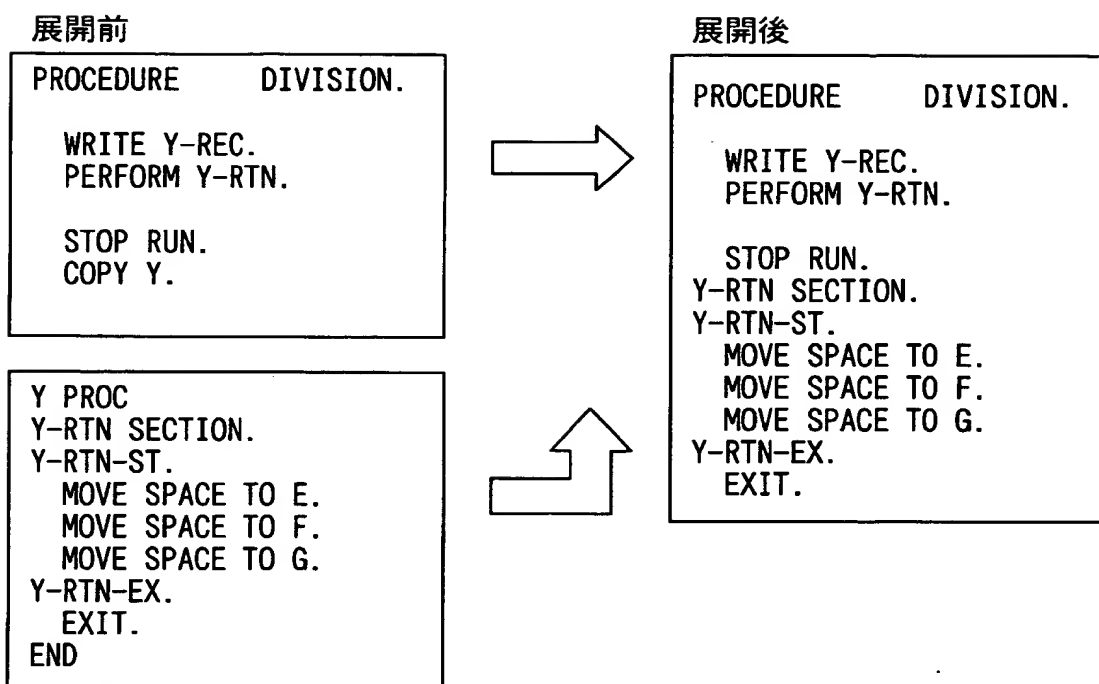
第 4 3 図



THIS PAGE BLANK (USPTO)

44 / 50

第44図



THIS PAGE BLANK (USPTO)

45 / 50

第45図

展開前

展開後

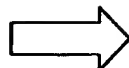
メインプログラム

メインプログラム

```

WORKING-STORAGE SECTION.
01 WK-A          PIC 9(10).
01 WK-B          PIC 9(10).
01 WK-C          PIC 9(10).
    ....
PROCEDURE         DIVISION.
    :
    :
    MOVE A TO WK-A.
    MOVE B TO WK-B.
    CALL 'SUB001'
        USING WK-A WK-B WK-C.
    MOVE WK-C TO C.
    :

```



```

WORKING-STORAGE SECTION.
01 WK-A          PIC 9(10).
01 WK-B          PIC 9(10).
01 WK-C          PIC 9(10).
    ....
PROCEDURE         DIVISION.
    :
    :
    MOVE A TO WK-A.
    MOVE B TO WK-B.
    WK-C=2 * WK-A * WK-B * WK-B
    :
    MOVE WK-C TO C.
    :
    :

```

サブルーチン

```

PROGRAM-ID.          SUB001.
    . . . .
LINKAGE SECTION.
01 WK-X              PIC 9(10).
01 WK-Y              PIC 9(10).
01 WK-Z              PIC 9(10).
PROCEDURE            DIVISION.
    USING WK-X WK-Y WK-Z.
PROC-RTN.            SECTION.
PROC-ST.

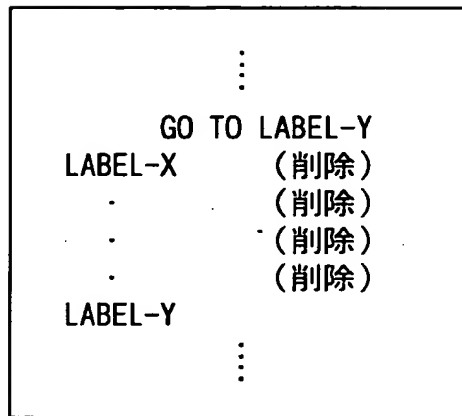
    COMPUTE
        WK-Z=2 * WK-X * WK-Y * WK-Y
PROC-EX.
    EXIT PROGRAM.

```

THIS PAGE BLANK (USPTO)

46 / 50

第46図



THIS PAGE BLANK (USPTO)

47/50

第47図

SET IDX TO 1 \Rightarrow MOVE 1 TO IDX

THIS PAGE BLANK (USPTO)

48 / 50

第 48 図

元の命令

整形後

MOVE A TO B,C ⇒ MOVE A TO B
MOVE A TO C

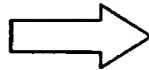
THIS PAGE BLANK (USPTO)

49/50

第49図

元の命令

```
IF  A = B
  AND C = D
  IF  X = Y
    AND Z = W
    MOVE E TO F
  ELSE
    MOVE G TO H
```



整形後

```
IF A = B AND C = D
IF X = Y AND Z = W
MOVE E TO F
ELSE
MOVE G TO H
END-IF
ELSE
END-IF
```

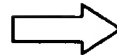
THIS PAGE BLANK (USPTO)

50 / 50

第50図

単階層化前

```
IF A = B
  PERFORM S1
ELSE
  PERFORM S2.
  ⋮
S1 SECTION
  S1 の内容
EXIT
  ⋮
S2 SECTION
  S2 の内容
EXIT
```



単階層化後

```
IF A = B
  S1 の内容
ELSE
  S2 の内容
```

THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP00/07916

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl.⁷ G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl.⁷ G06F9/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1926-1996	Jitsuyo Shinan Toroku Koho	1996-2001
Kokai Jitsuyo Shinan Koho	1971-2001	Toroku Jitsuyo Shinan Koho	1994-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO, 98/19232, A1 (Fumio NEGORO), 07 May, 1998 (07.05.98) & EP, 947916, A1 & CN, 1235682, A	1-39
A	WO, 99/49387, A1 (The Institute of Computer Based Software Methodology and Technology), 30 September, 1999 (30.09.99) (Family: none)	1-39
A	JP, 2-10429, A (Mitsubishi Electric Corporation), 16 January, 1990 (16.01.90) (Family: none)	1-39

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:
 "A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

Date of the actual completion of the international search
06 February, 2001 (06.02.01)Date of mailing of the international search report
13 February, 2001 (13.02.01)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

THIS PAGE BLANK (USPTO)

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl⁷ G06F9/44

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl⁷ G06F9/44

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1926年-1996年

日本国公開実用新案公報 1971年-2001年

日本国実用新案登録公報 1996年-2001年

日本国登録実用新案公報 1994年-2001年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	WO, 98/19232, A1 (根来文生), 7. 5月. 1998 (07. 05. 98) & EP, 947916, A1 & CN, 1235682, A	1-39
A	WO, 99/49387, A1 (株式会社アイエスデー研究所), 30. 9月. 1999 (30. 09. 99) (ファミリーなし)	1-39
A	JP, 2-10429, A (三菱電機株式会社), 16. 1月. 1 990 (16. 01. 90) (ファミリーなし)	1-39

☐ C欄の続きにも文献が列举されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

06. 02. 01

国際調査報告の発送日

13.02.01

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

漆原 孝治

5B

9366

電話番号 03-3581-1101 内線 3546

THIS PAGE BLANK (USPTO)